

Deep Learning

Clustering and Manifold Learning



Dr Chris Willcocks
Department of Computer Science

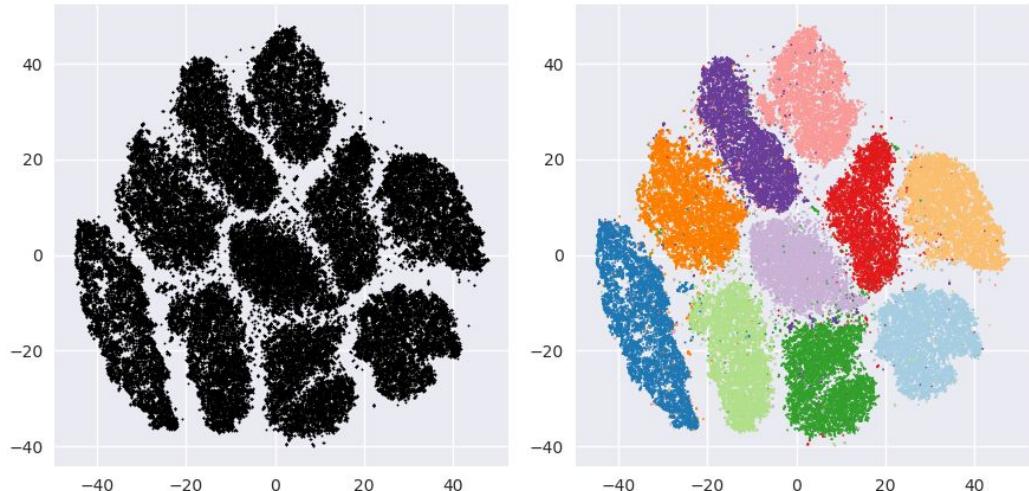
Lecture Overview

Recap

- Reinforcement Learning, RNNs
- Density Estimation

Today's lecture

- Clustering
 - Theory
 - k-Means
 - DBSCAN
 - ...
- Manifold Learning
 - T-SNE, UMAP
 - AE Embeddings, ...



Relation to Density Estimation

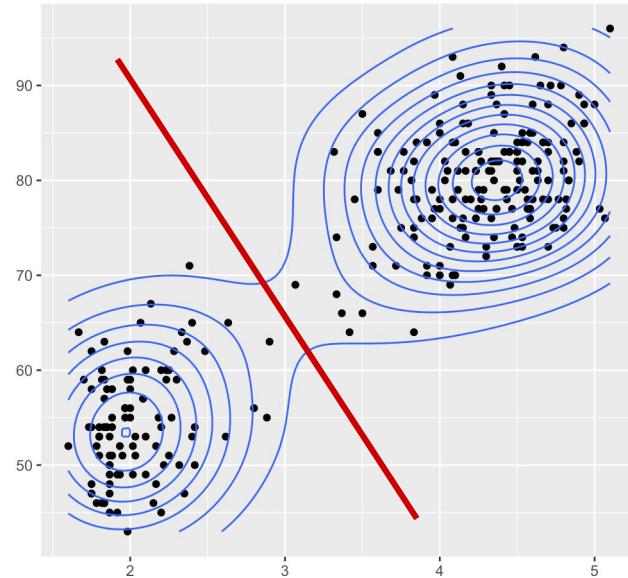
Manifold Learning

- Given some unorganised (high-dimensional) data can we learn some (low-dimensional) structure?



Clustering

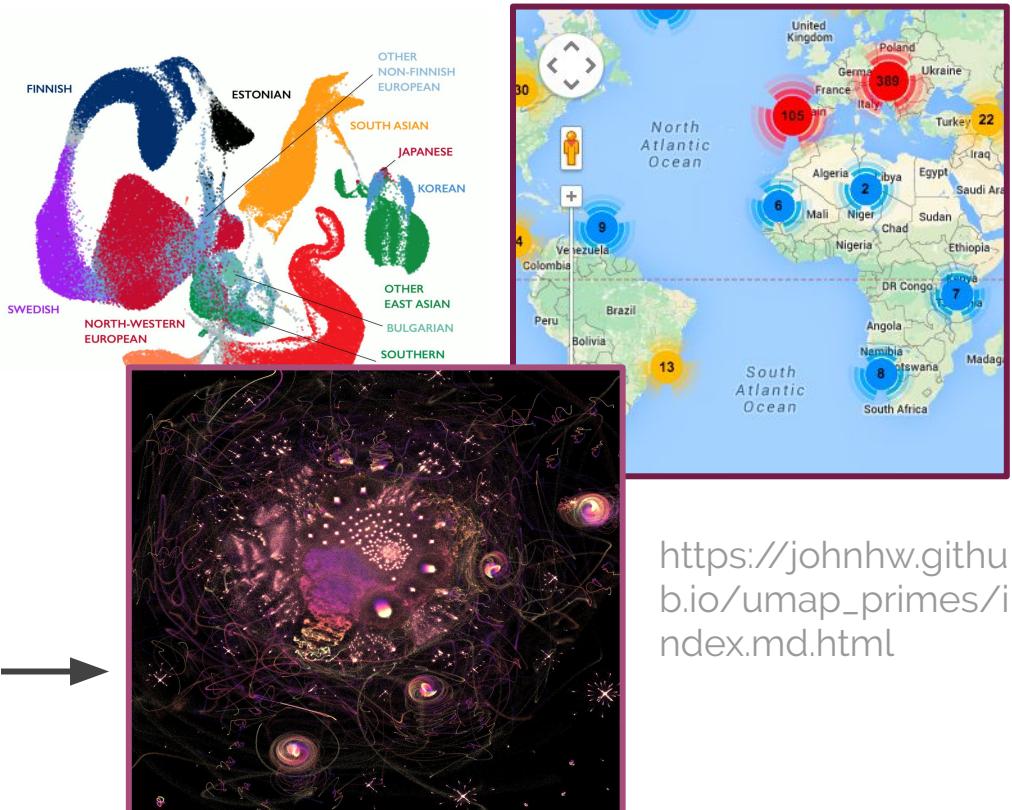
- Given some (structured) data, can we learn the decision boundaries?



Manifold Learning & Clustering Applications

- Plant and animal ecology
- Transcriptomics
- Business and marketing
- Credit risk & insurance
- Medicine
- Crime analysis
- Education
- Politics and demographics
- Chemistry
- Climatology
- *Pointless but beautiful math*
- ...

<https://macarthurlab.org/2018/10/17/gnomad-v2-1/>



https://johnhw.github.io/umap_primes/index.md.html

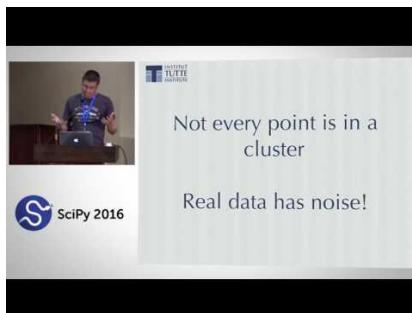
Clustering

The task

- Assign observations of a PDF to coherent clusters or label them as noise

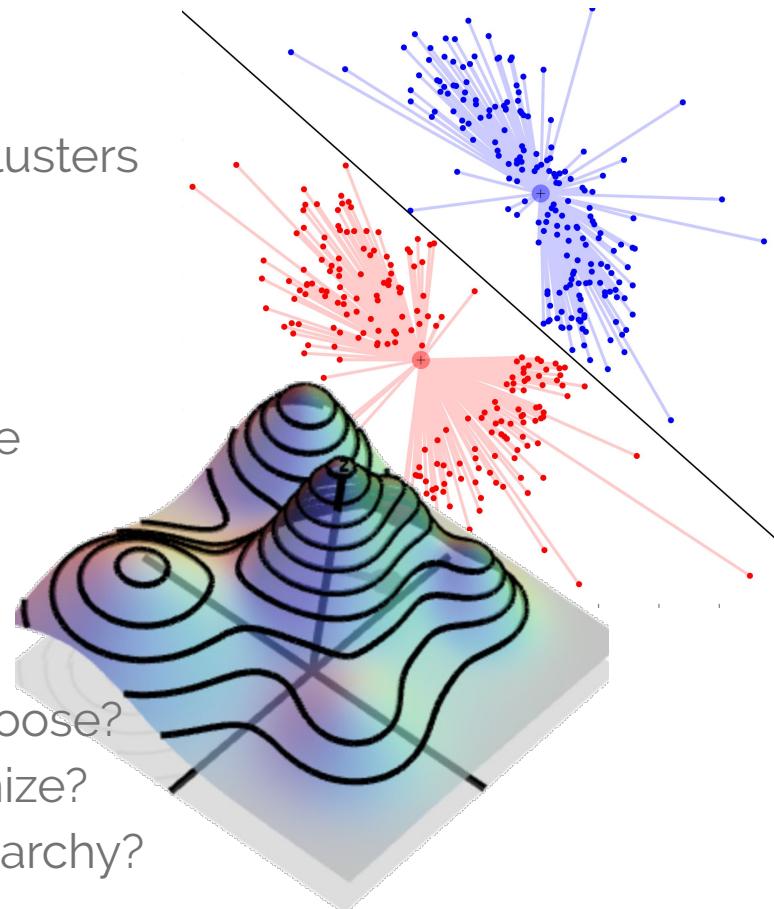
A **cluster** is a...

- **Connected-component** of a **level set** of the **unknown PDF** over our data **observations**



Some of the challenges

- What algorithm to choose?
- What metric to minimize?
- What scale, what hierarchy?



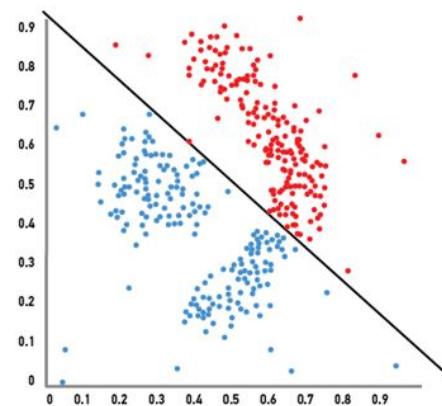
Types of Clustering Algorithm

Assignment:

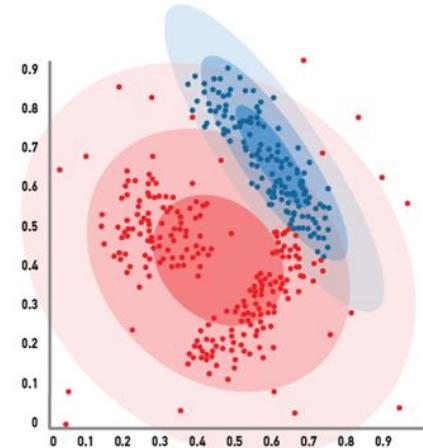
1. **Hard** clustering
2. **Soft** (fuzzy) clustering

Categories

- Centroid-based
- Density-based
- Distribution-based
- Connectivity-based
 - Bottom-up
 - Top-down

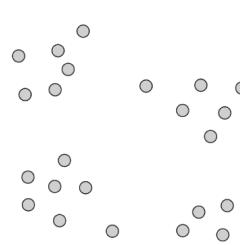


Hard clustering

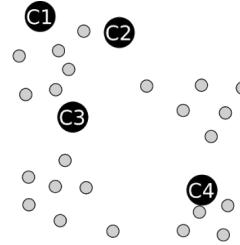


Fuzzy clustering

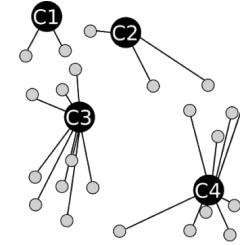
k-Means Clustering Algorithm



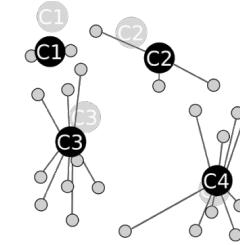
Input data



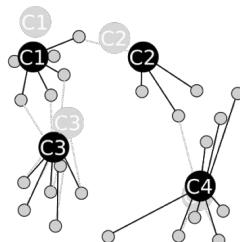
(1) Pick k
random
centroids



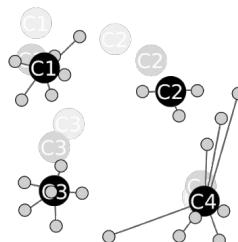
(2) Assign each
point to nearest
centroid



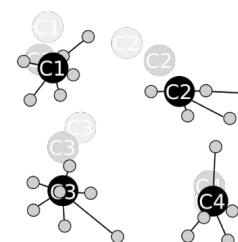
(3) Update
centroids to mean
of assigned



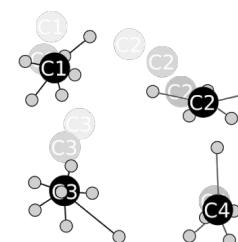
Repeat (2)



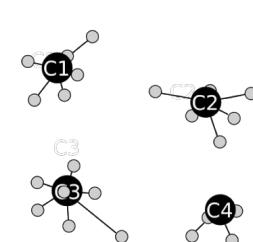
Repeat (3)



Repeat (2)

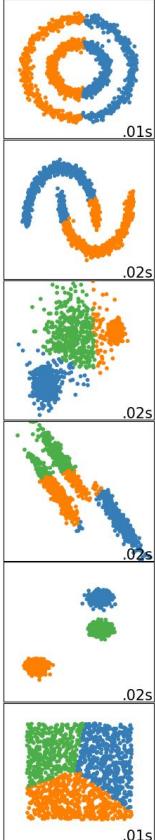


Repeat (3)

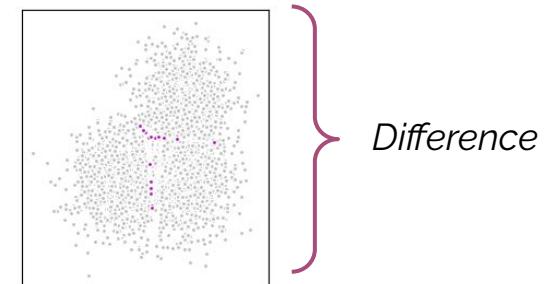
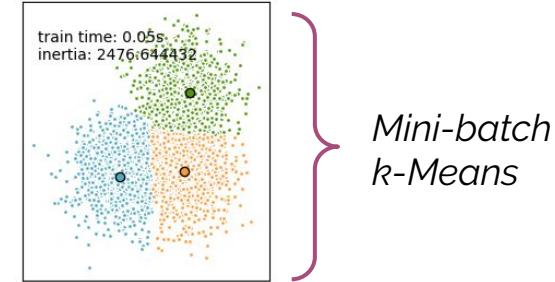
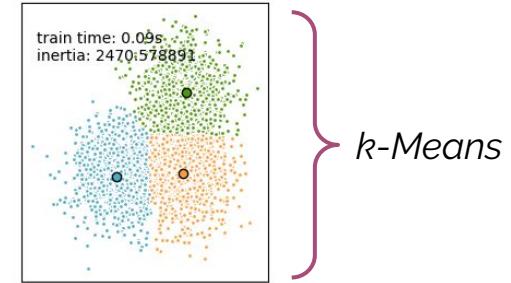
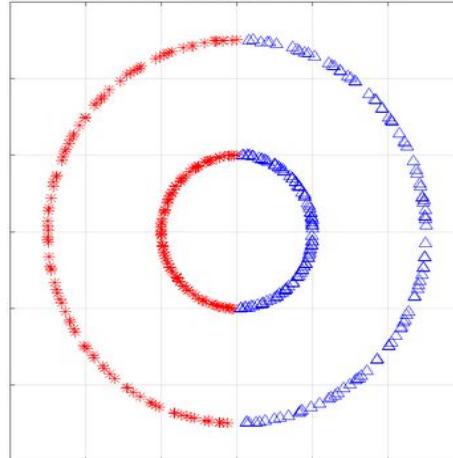
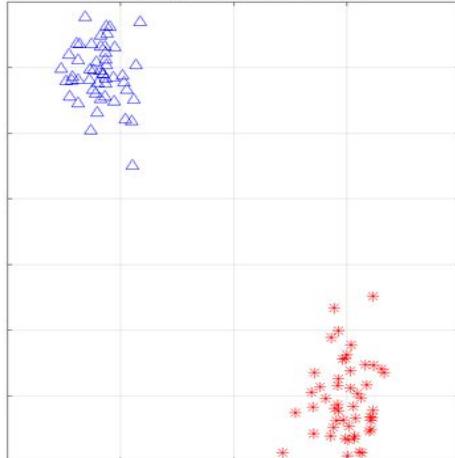


No movement =
convergence

Properties (limitations) of k -Means



- Requires k to be known in advance (a prior)
- Assigns noise to clusters
- Can exhibit slow convergence
 - Mini-batch k -means, spatial index data structure
- Requires data to be linearly separable (convex clusters):



*Mini-batch
 k -Means*

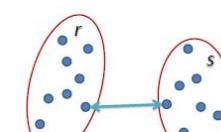
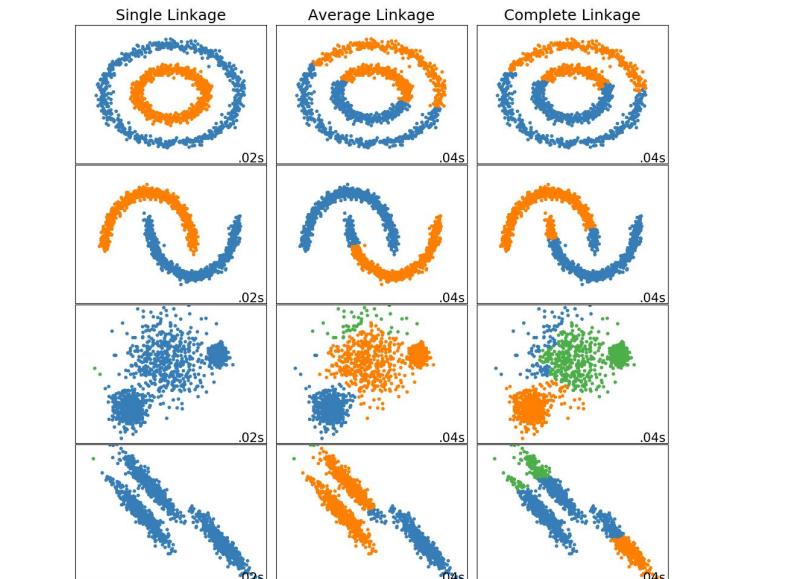
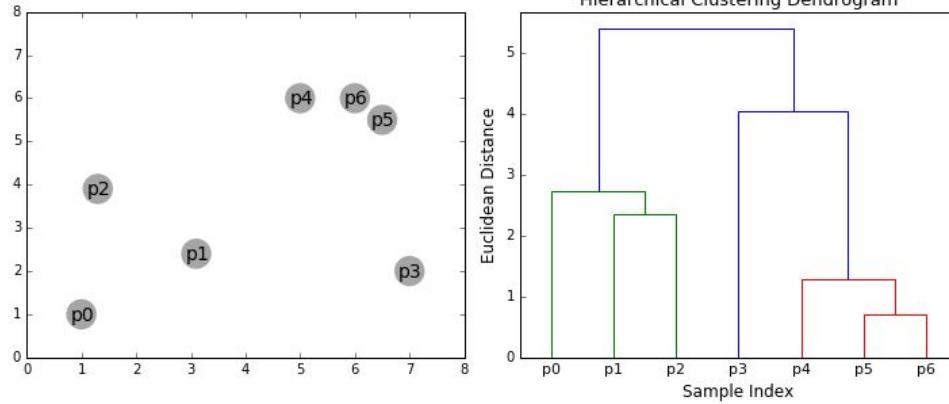
Difference

Hierarchical Clustering

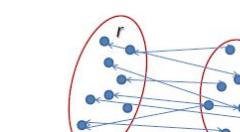
Hierarchical clustering methods are:

1. **Top-down** (divisive)
2. **Bottom-up** (agglomerative)

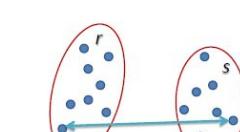
Various splitting/merging strategies....



$$L(r,s) = \min(D(x_{ri}, x_{sj}))$$



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj})$$



$$L(r,s) = \max(D(x_{ri}, x_{sj}))$$

Density-based (DB) Clustering

A Density-Based Algorithm for Discovering Clusters in Large Spatial

<https://www.aaai.org/Papers/KDD/1996/KDD96-037.pdf> ▾

by M Ester - Cited by 13952 - Related articles

In this [paper](#), we present the new clustering algorithm **DBSCAN** relying on density-based notion of clusters which is designed to discover clusters of

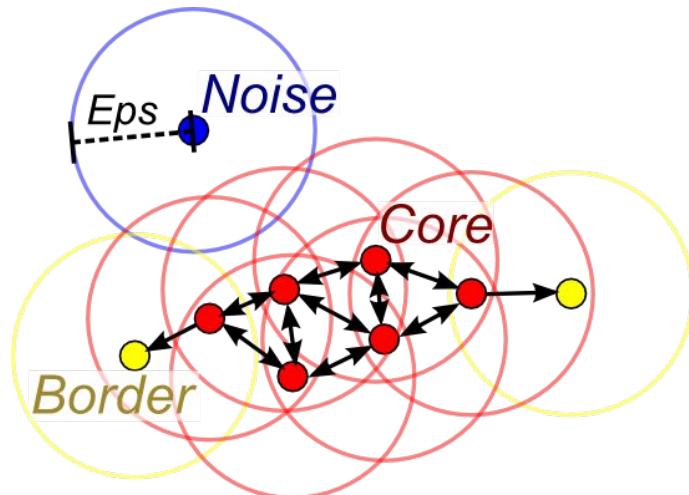
DBSCAN: Density-Based Spatial Clustering of Applications with Noise

Two parameters:

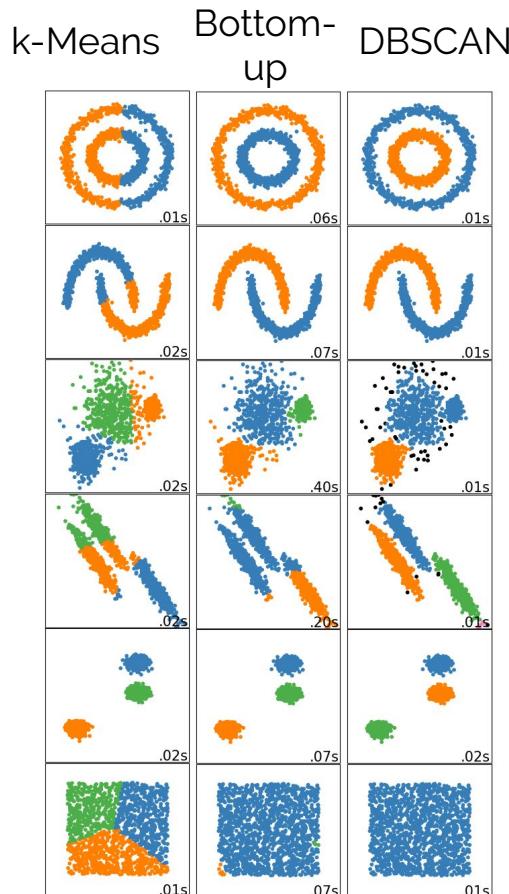
- Maximum radius $\epsilon = 0.4$
- Minimum points $\delta = 5$

Algorithm:

1. For all points, identify core points
2. Find core points connected components,
these are the clusters
3. Assign density-reachable borders also to clusters, otherwise its noise



Properties of DBSCAN

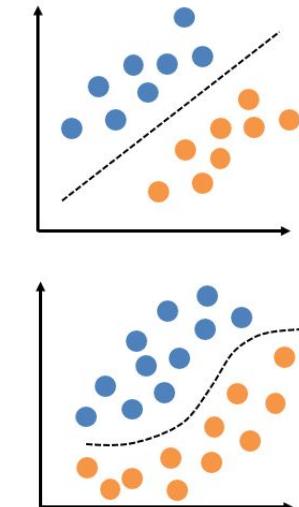


Main strengths

- Creates arbitrarily shaped clusters
- Classifies noise
- Average runtime $n \log n$ with spatial indexing structure

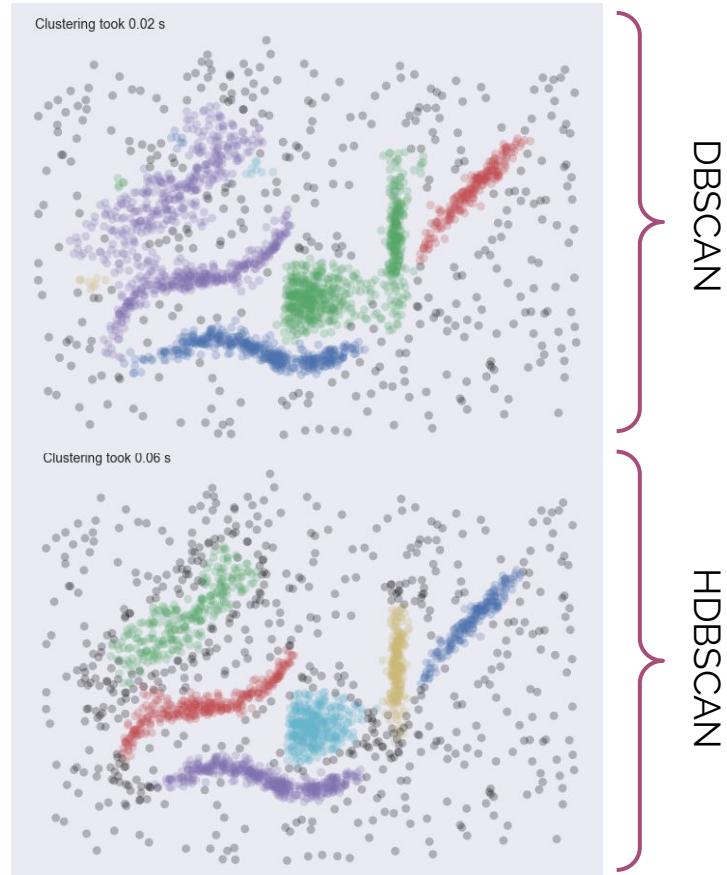
Main limitations

- Quite difficult to tune ε and δ
- Non-deterministic
 - Although modification exists
- Relies on euclidean distance measure



HDBSCAN (Hierarchical DBSCAN)

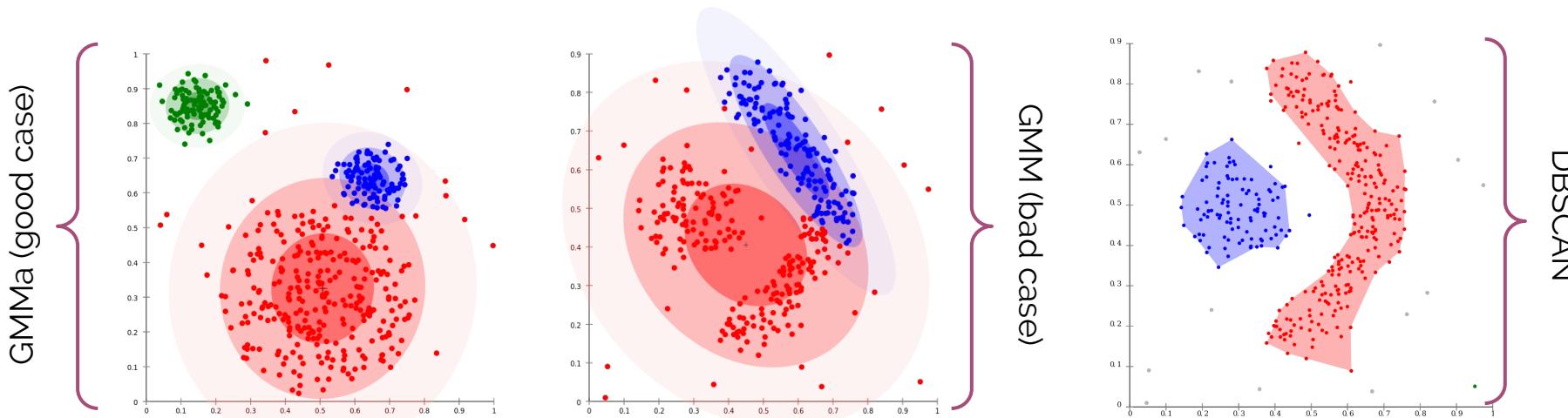
- More recent algorithm by some of same authors of DBSCAN
- Combines Hierarchical clustering with DBSCAN
- Just one intuitive parameter:
 - **minClusterSize**
- (Also has a *minSamples* parameter but is insensitive to it and can be set with sane defaults)
- Supports clusters of varying density
- Can be implemented reasonably efficiently



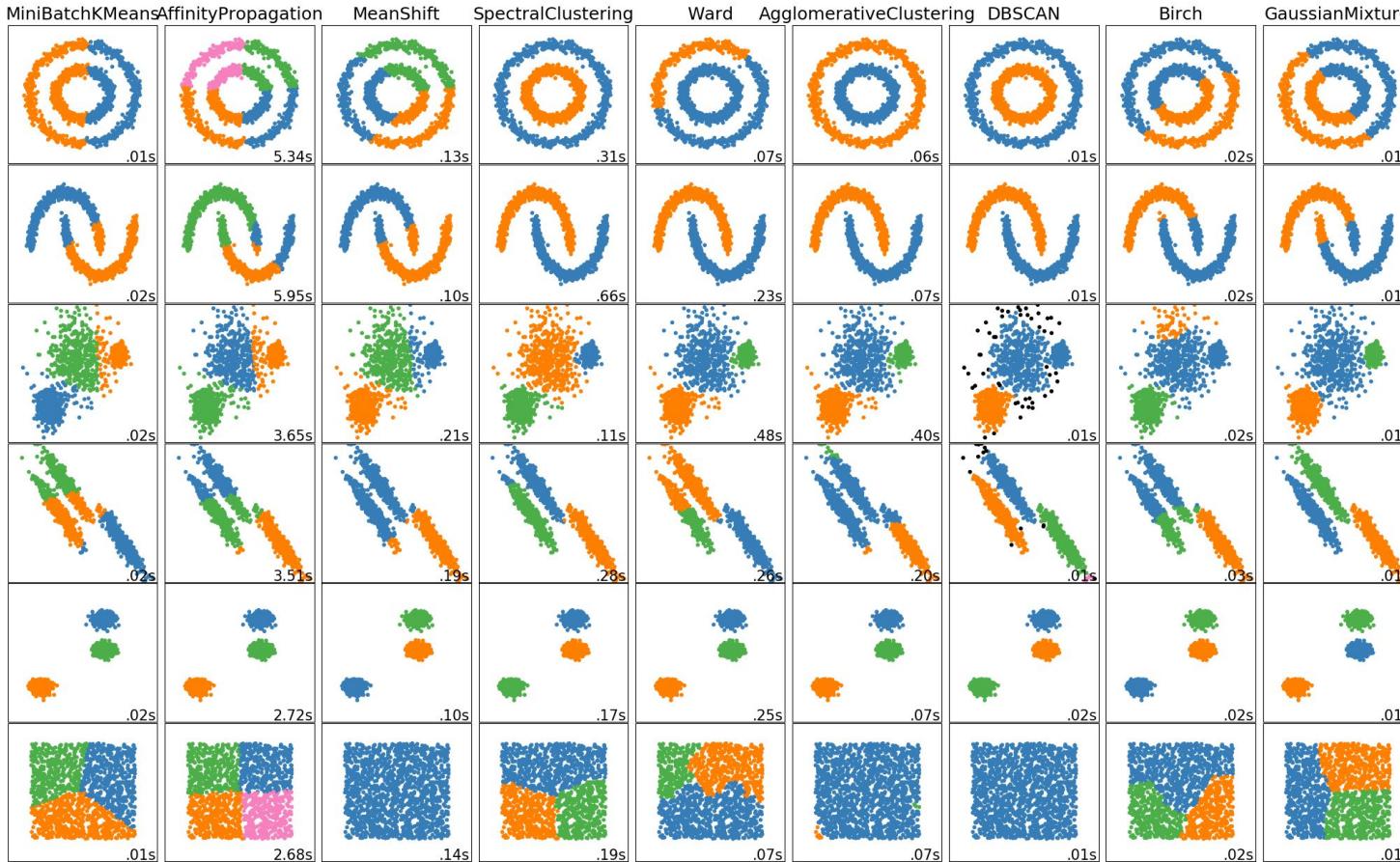
Distribution-based Clustering

Supposing our data points are generated from a mixture of a finite number of Gaussian distributions with unknown parameters, we can...

- Assign to distribution in a k-Means like fashion
- Optimise the parameters of the prior distributions such as to maximize the likelihood of the data given those assignments. Then repeat.



Clustering Algorithms...



Manifold Learning (or NLDR)

The **task**

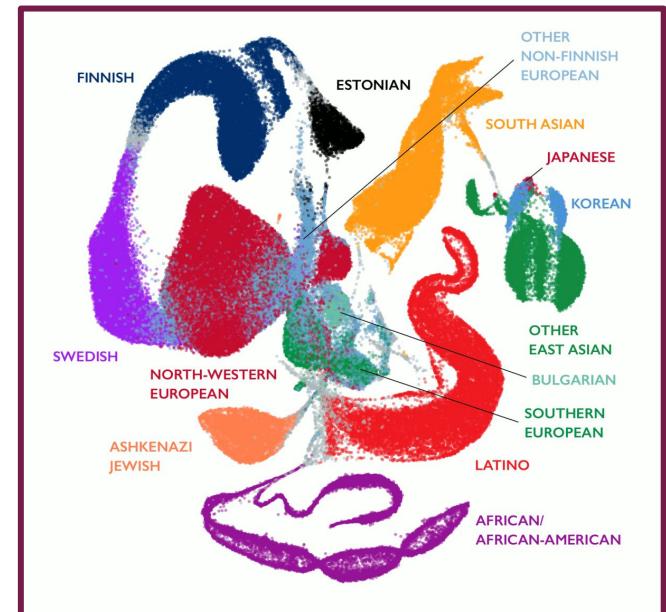
- Given some unorganised (high-dimensional) data can we learn some organised (low-dimensional) manifold that captures the data distribution?

The **challenges**

- What should the manifold look like, and why?
- Curse of dimensionality
 - Distances in high-dimensional spaces are all similar
- What neighborhood to define?
- What metric to minimize?

The **application:**

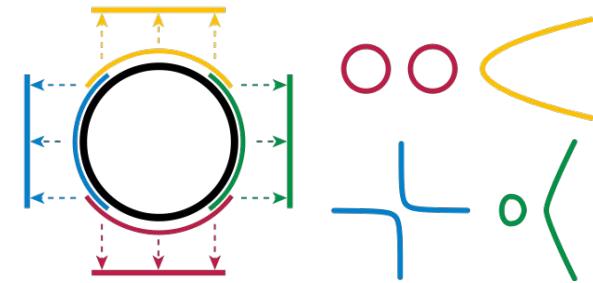
- Save space, understand data patterns, ...



Terminology

Manifold

- A space that locally resembles Euclidean space.
- Hierarchy of types of manifold:
 - Topological
 - Differentiable
 - Riemannian



Embedding

- An injective map $\phi : \mathcal{M} \rightarrow \mathcal{N}$ that preserves the underlying structure.

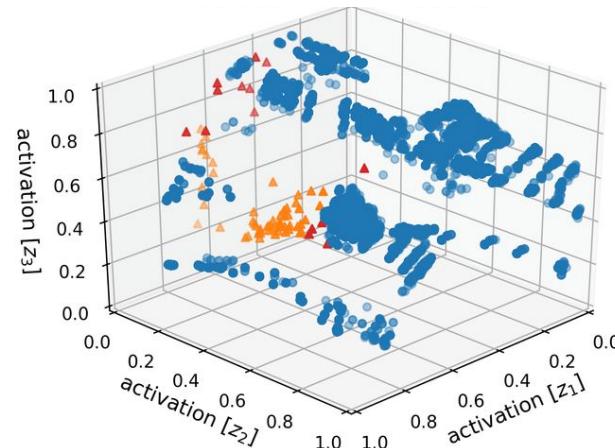
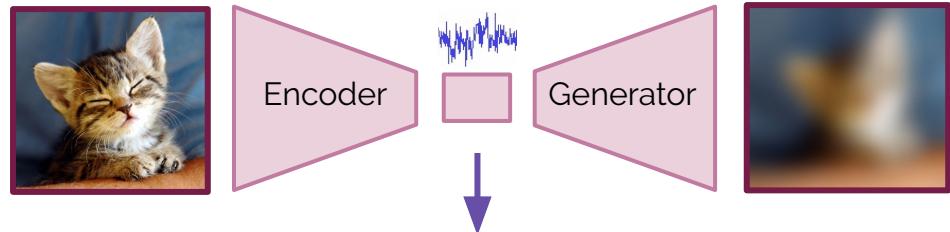
“Recovering the riemannian metric”



Autoencoders

Learn a non-linear mapping from high-dimensional to an embedded space

- Local variations with L_2
- **Can fracture** a manifold into many different domains (giving different regions for nearby data points)
- If we have **prior knowledge** about data, we can split the manifold into lovely categories easily
 - Global coherency



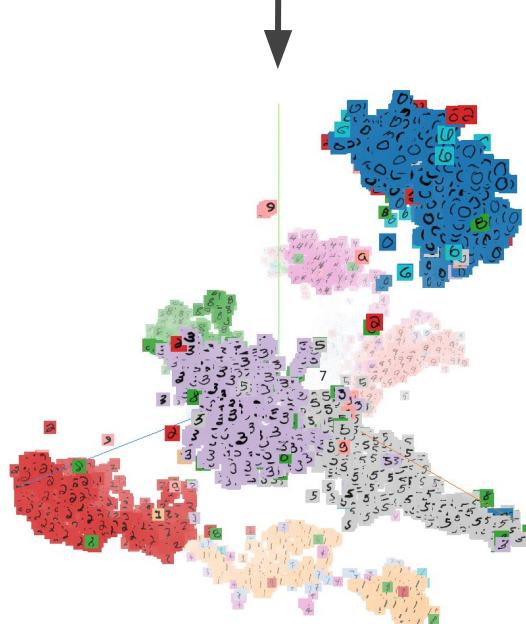
T-Distributed Stochastic Neighbor Embedding

The t-SNE algorithm:

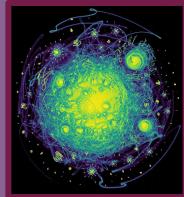
- Uses local relationships between points to create a low-dimensional mapping.
 - Defines this relation by assuming an **adaptive Gaussian prior** based on a “perplexity” hyperparameter
- Defines a distribution over points in low-dimensional space, and minimizes KL-divergence between the two distributions
 - Ensures dissimilar points are far apart

Visualizing Data using t-SNE - Journal of Machine Learning Research
www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf ▾
by L Maaten - 2008 - Cited by 7224 - Related articles
We present a new technique called “t-SNE” that visualizes high-dimensional data.... present in the paper are sufficient to demonstrate the superiority of t-SNE.

9 3 1 9 7 2 4 5 1 0 3 2 4 3 7 5 9 0 3 4 9 8 6 6 8 0 9 6 5 1 6 7 9 5 4
3 0 2 4 2 9 4 8 3 2 0 1 3 5 3 5 7 4 6 8 5 1 4 1 6 9 6 9 0 1 4 2 1 3 1
2 8 2 3 2 3 8 2 4 9 8 2 9 1 3 9 1 1 1 9 9 6 6 9 7 9 4 2 2 6 3 3 3 1 6
6 3 6 9 0 3 6 0 3 0 1 1 3 9 3 1 5 0 4 9 6 8 7 1 0 3 7 9 9 1 8 1 7 2 2
3 3 8 0 7 0 5 6 9 8 2 4 1 4 4 4 2 4 9 5 3 3 4 8 2 4 2 0 4 3 2 6 1 4 0 6 3
1 1 9 5 8 0 4 3 7 7 5 0 5 4 2 0 9 8 1 2 4 9 3 5 2 0 0 5 1 9 3 9 6 1 8
9 5 0 0 5 1 1 1 7 4 7 7 2 6 5 1 8 2 4 1 1 5 6 5 7 3 3 0 4 3 8 5 4 6 7
0 7 1 6 1 7 0 9 5 6 3 2 6 6 7 1 5 2 3 2 3 5 6 8 5 0 2 0 2 7 9 2 4 6



UMAP



UMAP: Uniform Manifold Approximation and Projection for ...

by L McInnes - 2018 - Cited by 631 - Related articles

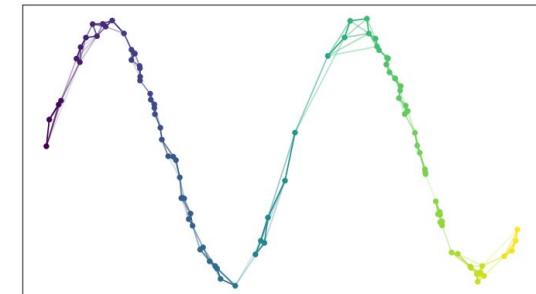
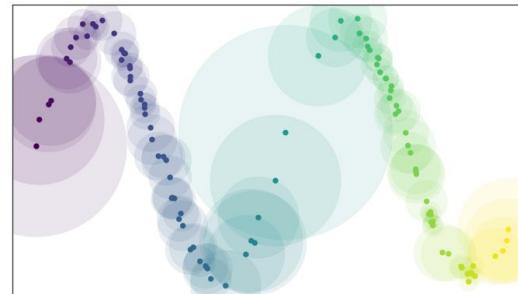
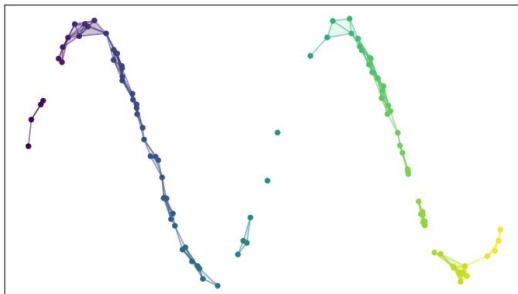
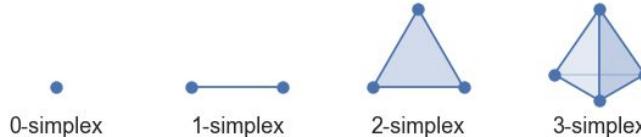
9 Feb 2018 - Abstract: **UMAP** (**Uniform Manifold Approximation and Projection**) is a manifold learning technique for dimension reduction. **UMAP** is ...

The UMAP algorithm:

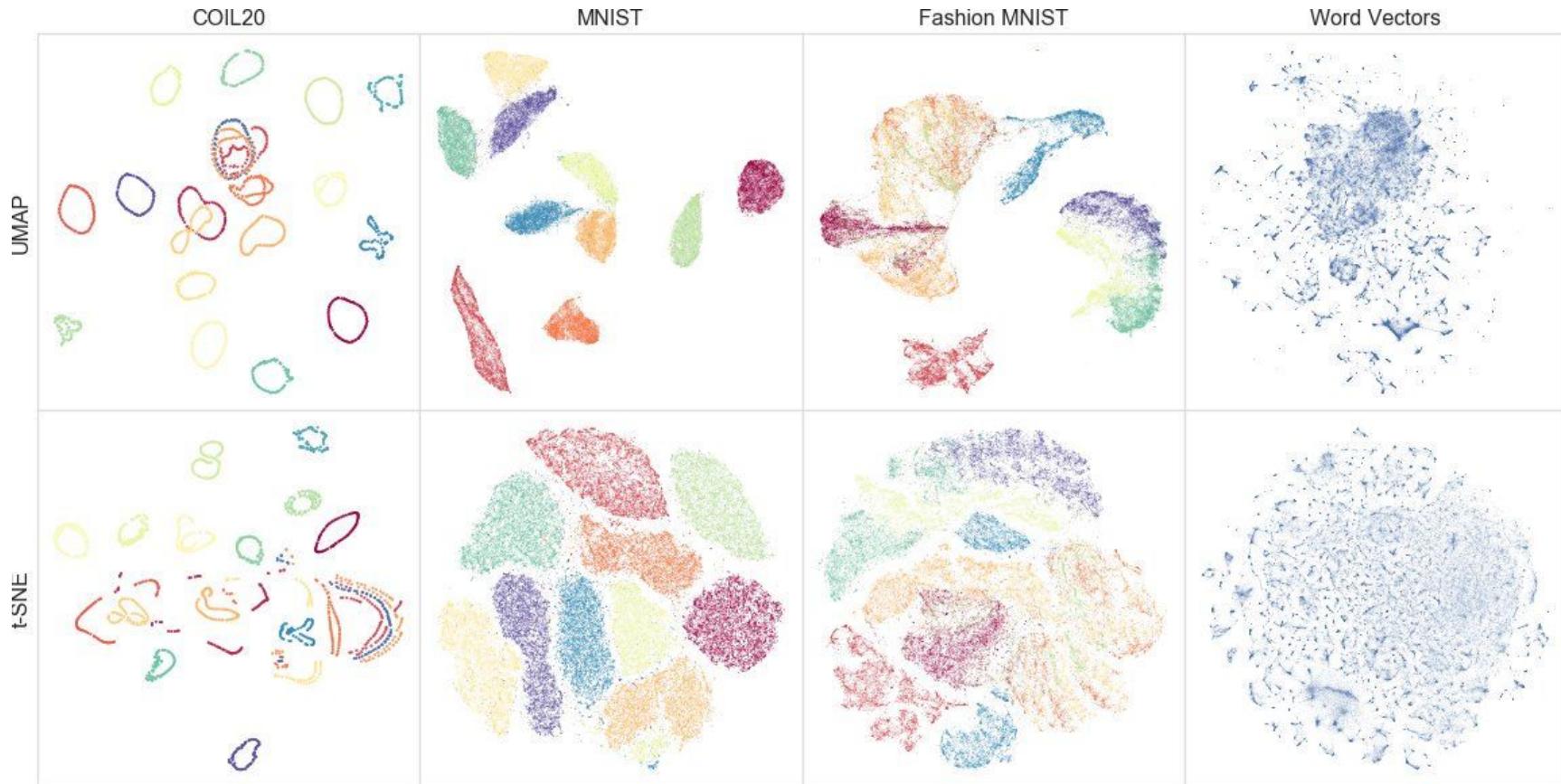
- Finds k-nearest neighbors in dataset
 - Creates edge weights based on diameter of neighborhood graph and distance to nearest neighbor
- Minimize:

$$\sum_{e \in E} w_h(e) \log \left(\frac{w_h(e)}{w_l(e)} \right) + (1 - w_h(e)) \log \left(\frac{1 - w_h(e)}{1 - w_l(e)} \right)$$

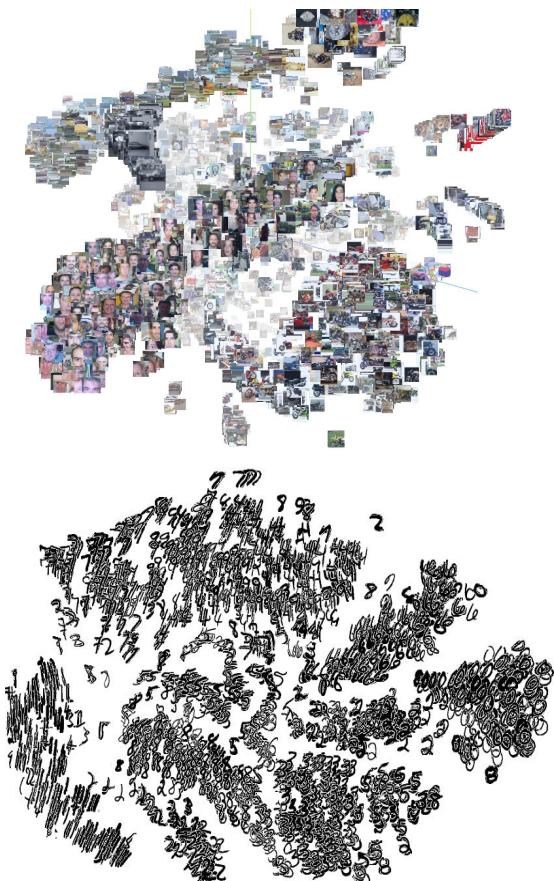
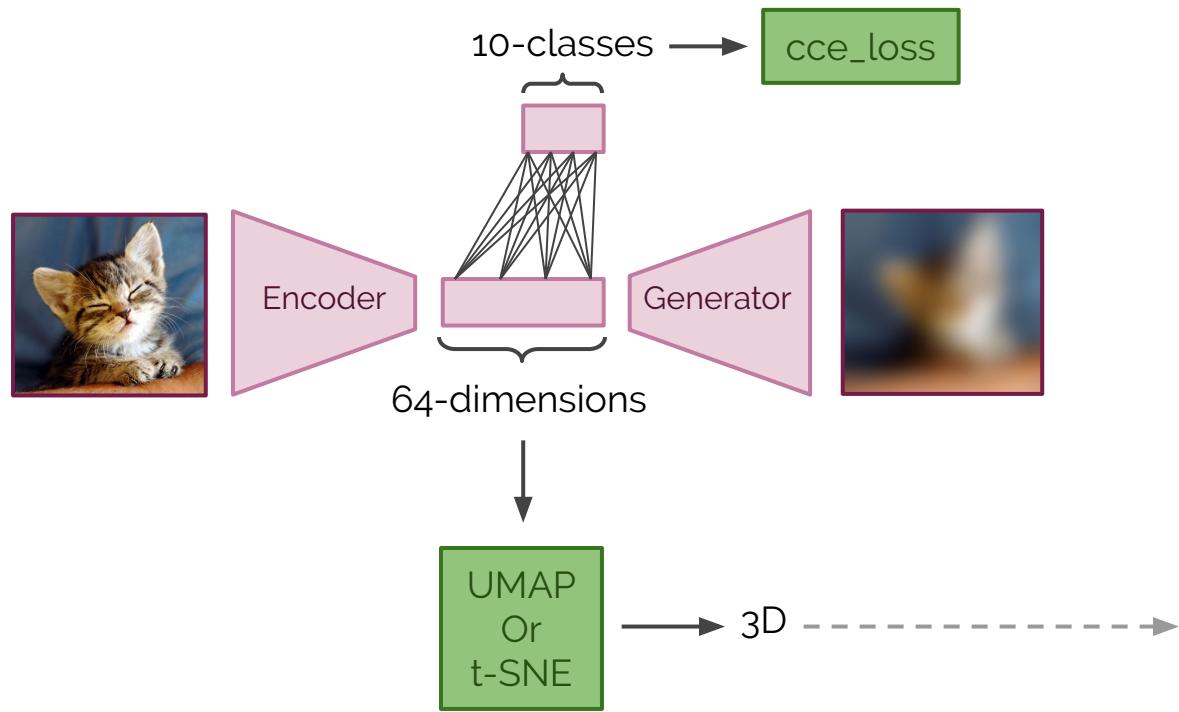
Get the clumps right Get the gaps right



UMAP and t-SNE Examples

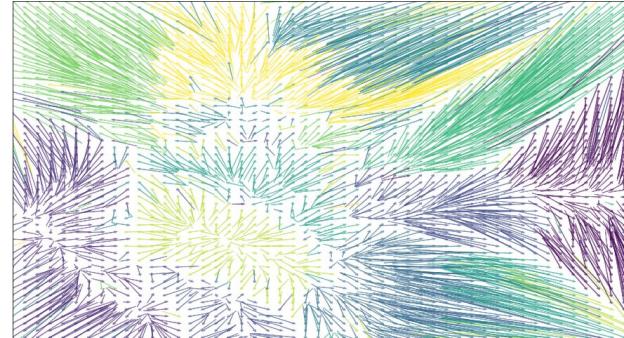
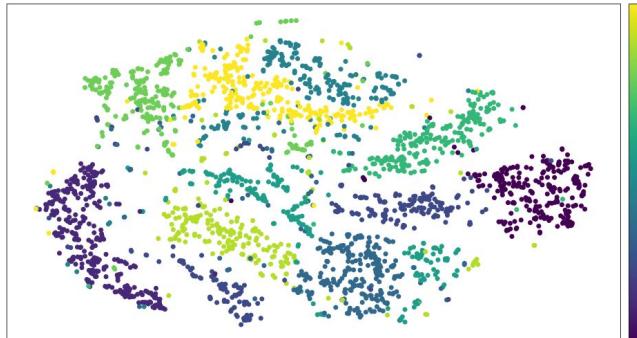


Learning Tailored Manifolds



Take Away Points

- There is no one clustering algorithm that outperforms all...
 - But density-based stuff is generally pretty good in low-dimensions
- There's an infinite number of good manifolds out there
 - **UMAP** is currently a pretty good NLDR algorithm
- Good density estimators underpin a lot of this field
 - Especially defining local neighborhoods is important
- In practice, UMAP works well with **HDBSCAN**.
 - See: <https://umap-learn.readthedocs.io/en/latest/clustering.html>



Jonker-Volgenant
algorithm