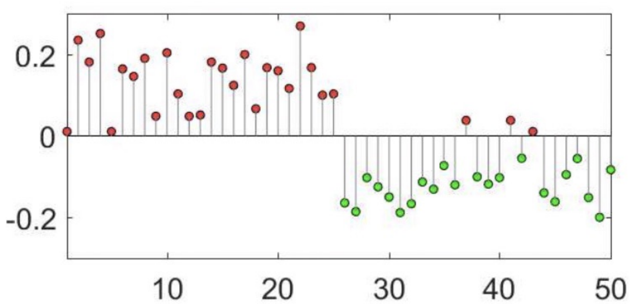
LECTURE 32

· Last class: community detection in SBM $G(n,p,q)$. Spectral Clustering:

Second top eigenvector $v_2(A)$ of the adjacency matrix A

coefficients of $v_2(A)$ cluster, predict communities:



We proved guarantees.

If $p = \frac{a}{n}$, $q = \frac{b}{n}$, $(a-b)^2 > C(a+b)\log n$,

99% accuracy with 99% probability.

① Other approaches to community detection for SBM:

Instead of the adjacency matrix A, one can use:

(a) graph Laplacian $\mathcal{L} = I - D^{-1/2} A D^{-1/2}$ [Le-Levina-V 2017]

(b) non-backtracking matrix [Mossel-Neeman-Sly 2018, Massoulie 2015]

(c) semidefinite relaxations [Abbe-Bandeira-Hall 2015, Guedon-V 2016]

Advantage: removes log factor. Guarantees:

- If $(a-b)^2 < 2(a+b)$, no method works [MNS]
- If $(a+b)^2 > 2(a+b)$, >50% accuracy [MNS, M] (better than a random guess)
- If $(a+b)^2 > C(a+b)$, 99% accuracy [MNS, LLV, GV]
- If $\sqrt{a} - \sqrt{b} > 2\sqrt{\log n}$, 100% accuracy [ABH]

SKIP

② Methods based on semidefinite relaxation:

- Want to maximize #(edges within communities) − #(edges across communities)
- Let $x \in \{\pm 1\}^n$ encode community membership $\Rightarrow$

here $x_i = x_j \Rightarrow x_i x_j = 1$ here $x_i = -x_j \Rightarrow x_i x_j = -1$

communities have same size $n/2$

$$\Rightarrow \boxed{\max\Big\{ \underbrace{\sum_{i,j=1}^n A_{ij} x_i x_j}_{= \langle A,\, xx^T \rangle} : \underbrace{x_i = \pm 1}_{\Updownarrow \; (xx^T)_{ii} = x_i^2 = 1},\ \underbrace{\sum_{i=1}^n x_i = 0}_{\Updownarrow \; \langle \mathbb{1},\, xx^T \rangle = \sum_{i,j=1}^n 1 \cdot x_i x_j = \left(\sum_{i=1}^n x_i\right)^2 = 0} \Big\}} \quad \textbf{NP-hard.}$$

Semidefinite relaxation?

$\mathbb{1}$: matrix whose all entries $=1$

$$\Rightarrow \text{SDP: } \boxed{\max\{\langle A, X\rangle : X \succeq 0;\ X_{ii} = 1\ \forall i;\ \langle \mathbb{1}, X\rangle = 0\}} \quad \textbf{Polynomial time.}$$

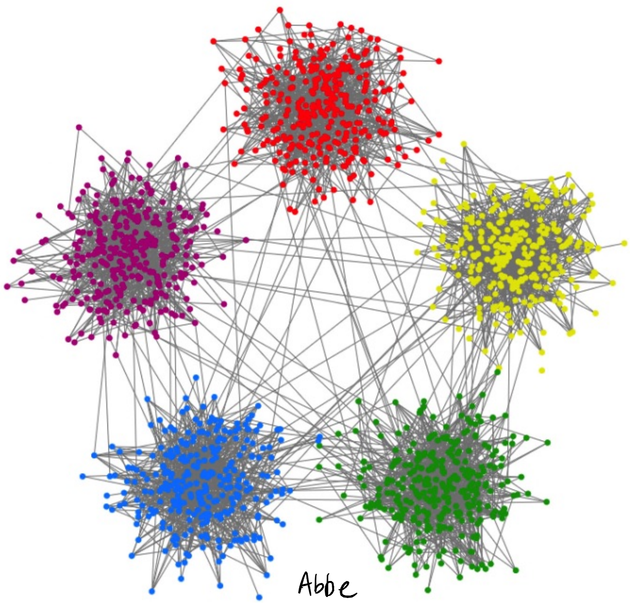
Theoretical guarantees:

- $(a-b)^2 \geq C(a+b) \Rightarrow$ 99% accuracy [Guédon-V 2016] (based on Grothendieck ineq.)
- $\sqrt{\frac{a}{\log n}} - \sqrt{\frac{b}{\log n}} > \sqrt{2} \Rightarrow$ 100% accuracy [Abbe-Bandeira-Hall '2015]

Disadvantages: slow.

② Beyond SBM

- More than 2 communities?



Include more eigenvectors of A, e.g. $k=3$:

$$\begin{array}{c|ccc|} & V_1 & V_2 & V_3 \\ \hline x_1 \to & & & \\ x_2 \to & & & \\ \vdots & & & \\ x_n \to & & & \\ \hline \end{array}$$

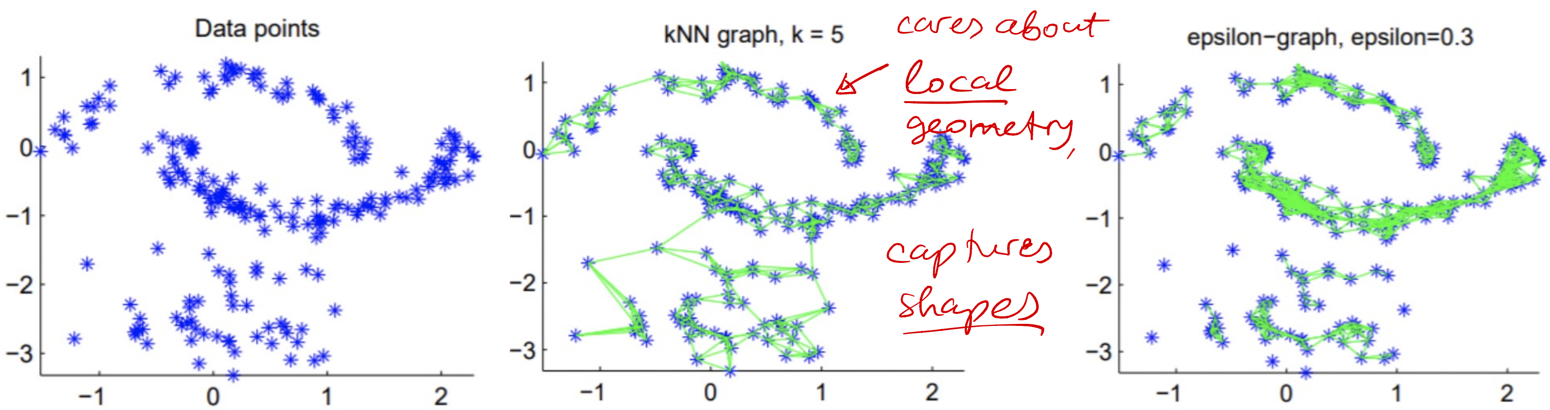
$\Rightarrow$ represent the graph G as the set of points $\{x_1, \dots, x_n\} \subset \mathbb{R}^k$.

- Cluster using e.g. K-means.
- or just visualize the graph

$\Rightarrow$ $\boxed{\text{Graph} \xrightarrow{\text{spectral embedding}} \mathbb{R}^k}$.

③ Vice versa: $\boxed{\text{data in } \mathbb{R}^d \longrightarrow \text{Graph}}$

- Data $\{x_1, \dots, x_n\} \subset \mathbb{R}^d \longrightarrow$ neighborhood graph:
  - $\to$ connect x_i, x_j if $\|x_i - x_j\| \leq \varepsilon$ (ε-graph)
  - $\to$ or connect x_i to k nearest neighbors (k-NN graph)



cares about local geometry,

captures shapes

②&③ $\Rightarrow$ **Graph theory $\longleftrightarrow$ Euclidean geometry**

Particularly: $x_1, \dots, x_n \in \mathbb{R}^d \longrightarrow$ graph $\longrightarrow$ $x_1, \dots, x_n \in \mathbb{R}^k$

neighborhood　　　　spectral embedding

Example [Kanevsky-Kaiser '2009]:

data $\subset \mathbb{R}^2$

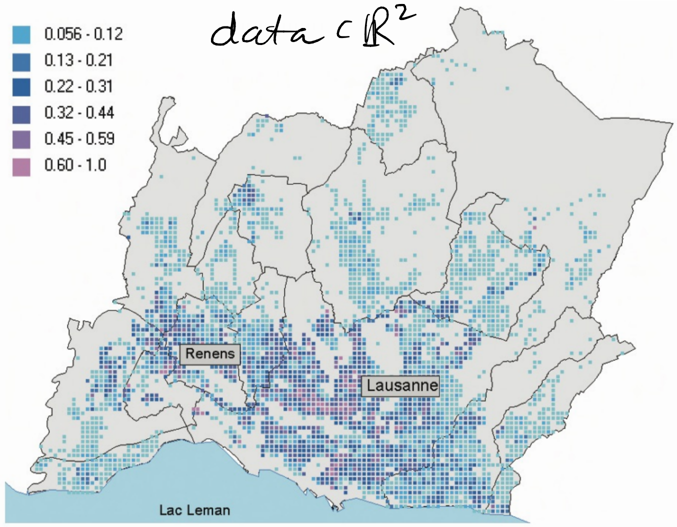


Figure 10: Population density in the region of Lausanne. The values are normalized to the maximum value of density in the region

data $\to$ k-means

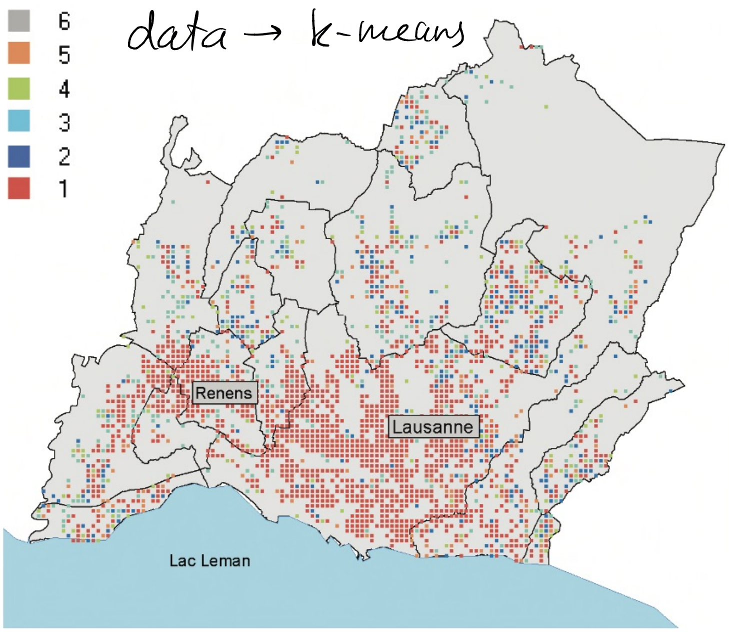


Figure 11: Clusters obtained with k-means method

data $\subset \mathbb{R}^2 \longrightarrow$ neighb. graph $\longrightarrow$ data $\subset \mathbb{R}^k$ $\downarrow$ k-means

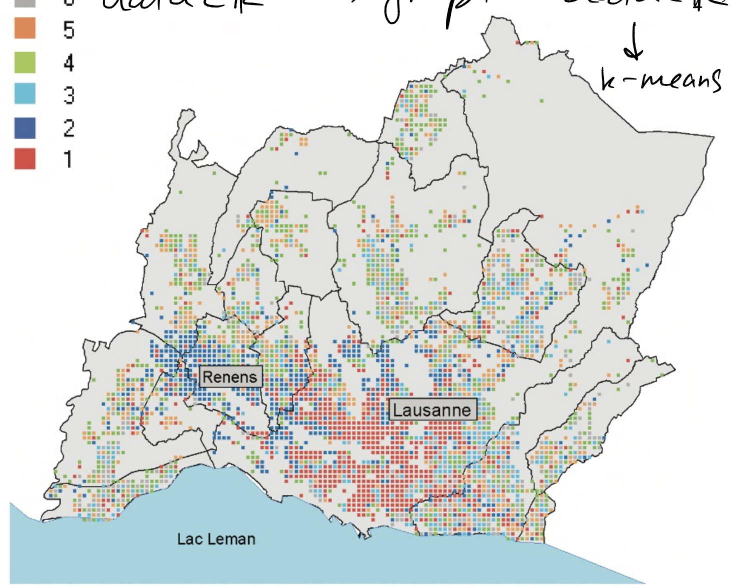


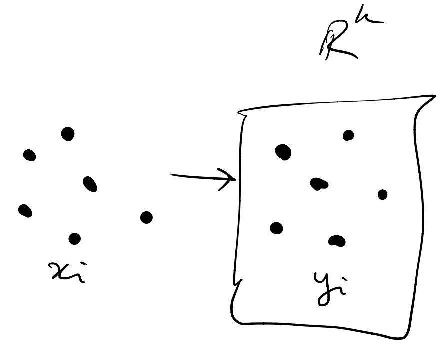
Figure 13: Clusters obtained with spectral clustering

+ k-means

④ An alternative to spectral embedding: multidimensional scaling (MDS)

Embeds $\forall$ finite metric space $(M, d) \hookrightarrow \mathbb{R}^k$

$M = \{x_1, \dots, x_n\}$, d ← metric



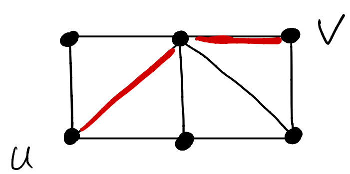
$$\min_{y_1, \dots, y_n \in \mathbb{R}^k} \sum_{i,j=1}^{n} \left(d(x_i, x_j) - \|y_i - y_j\|_2 \right)^2$$

- Pro: general.
  ↓
  In particular:
- Con: non-convex ($\Rightarrow$ no theory)

⑤ Graph as a metric space

- the "graph distance" between vertices = length of the shortest path.

Example:



$d(u, v) = 2$

- $\Rightarrow$ can use MDS to embed $\boxed{\text{graph} \xhookrightarrow{\text{MDS}} \mathbb{R}^k}$

Putting all together gives

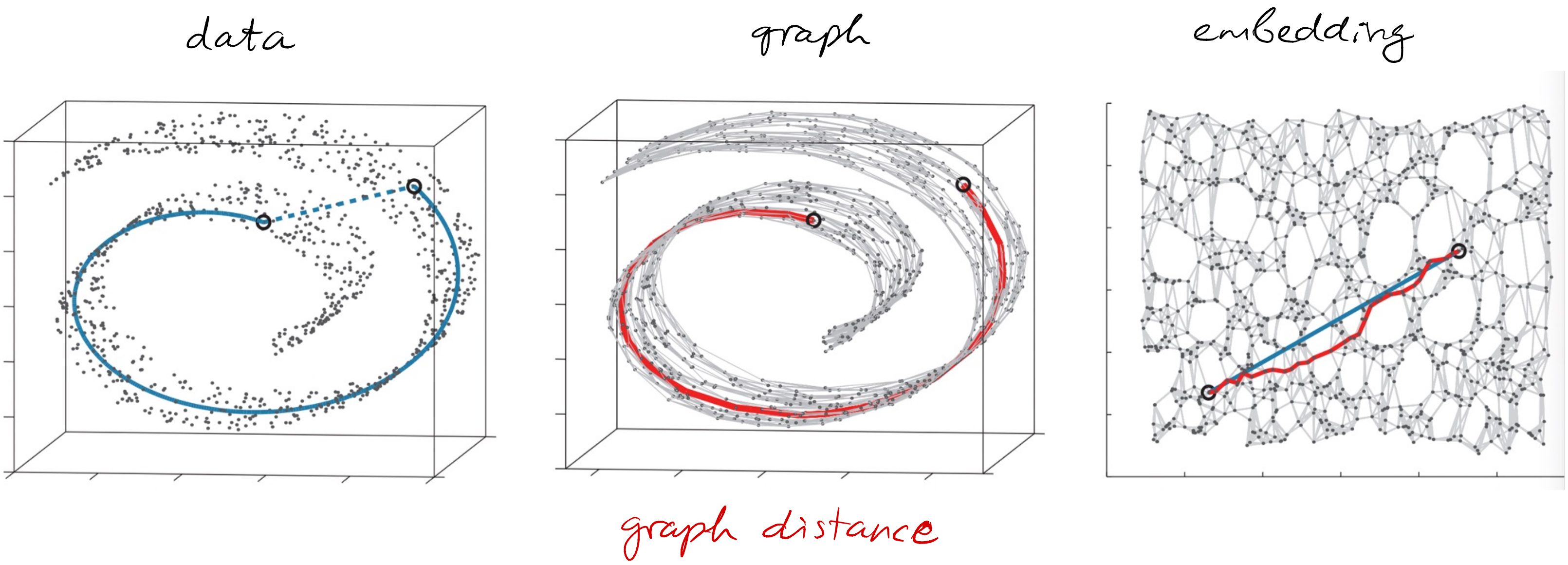
⑥ ISOMAP

<u>Input</u>: $x_1, \dots, x_n \in \mathbb{R}^d$

- construct a neighborhood graph (K-NN)
- compute the graph distance $d(x_i, x_j)$ $\forall i,j$
- apply MDS

<u>Output</u>: $y_1, \dots, y_n \in \mathbb{R}^k$ (e.g. $k=2$)

<u>Example</u>: Isomap unrolls a swiss roll [Tenenbaum et al. 2000]:

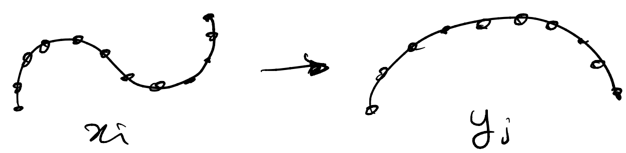


7. MDS + kernels = **SNE** (stochastic neighbor embedding):

Embed $x_1, \dots, x_n \in \mathbb{R}^n \longrightarrow y_1, \dots, y_n \in \mathbb{R}^k$ via

$$\min_{y_1,\dots,y_n \in \mathbb{R}^k} \sum_{i=1}^{n} \Big[\underbrace{\exp\Big(-\frac{\|x_i - x_j\|_2^2}{2\sigma^2}\Big)}_{p_{ij}} - \underbrace{\exp\Big(-\frac{\|y_i - y_j\|_2^2}{2\sigma^2}\Big)}_{q_{ij}} \Big]^2 . \qquad (*)$$

- Advantage: cares about <u>local</u> geometry, in σ



OK.

- <u>Probabilistic view</u>: consider p_{ij}, q_{ij} as probabilities (normalize them to have $\sum p_{ij} = \sum q_{ij} = 1$).

- Instead of the objective function $(*)$, minimize the KL divergence (a.k.a. relative entropy)

$$\boxed{KL(\underset{\text{data}}{P} \,\|\, \underset{\text{model}}{Q}) := \sum_{i,j=1}^{n} p_{ij} \log \frac{p_{ij}}{q_{ij}}}$$

- Heuristically, $KL(P\|Q)$ = amount of information in data (P) that is not explained by the model (Q)

Rigorously, Shannon thm:

$KL(P\|Q) = \mathbb{E}$ #bits to encode samples from P using code optimized for Q
$- \mathbb{E}$ #bits to encode samples from P using code optimized for P.

- KL objective function is more forgiving than sum-of-squares.

⑧ t-SNE

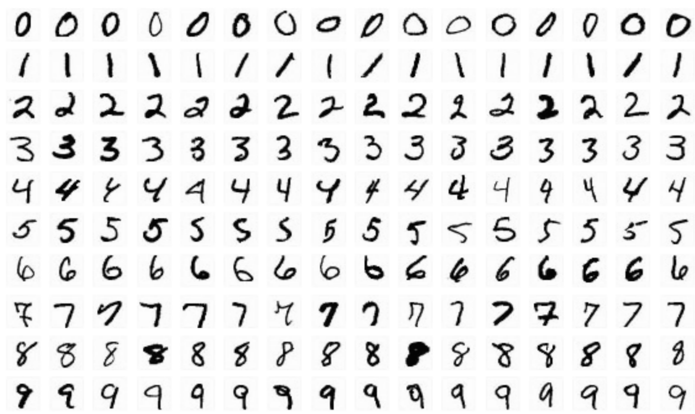
In the low-dimensional space, replace the Gaussian distribution by Cauchy distribution (more forgiving):

$$p_{ij} = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right), \qquad q_{ij} = \frac{1}{1+\|y_i - y_j\|_2^2}$$

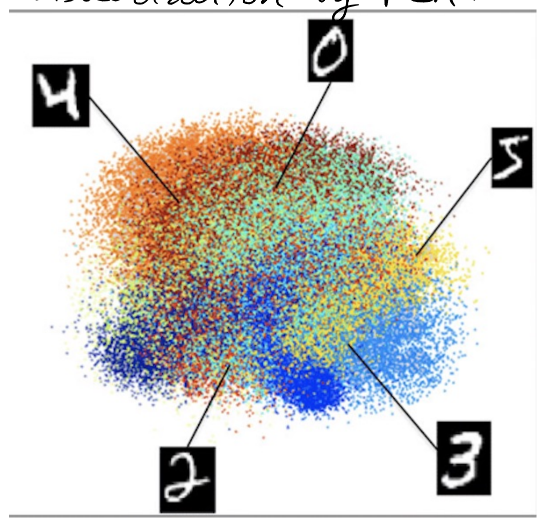
$$\min_{y_1,\dots,y_n} D(P\|Q) \Rightarrow \text{t-SNE}$$

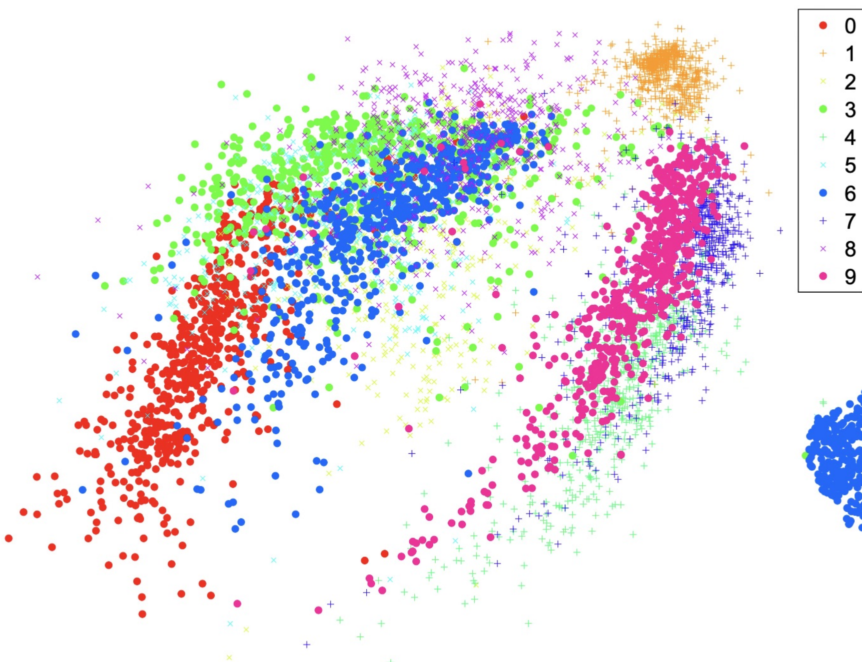
Example:

MNIST dataset:

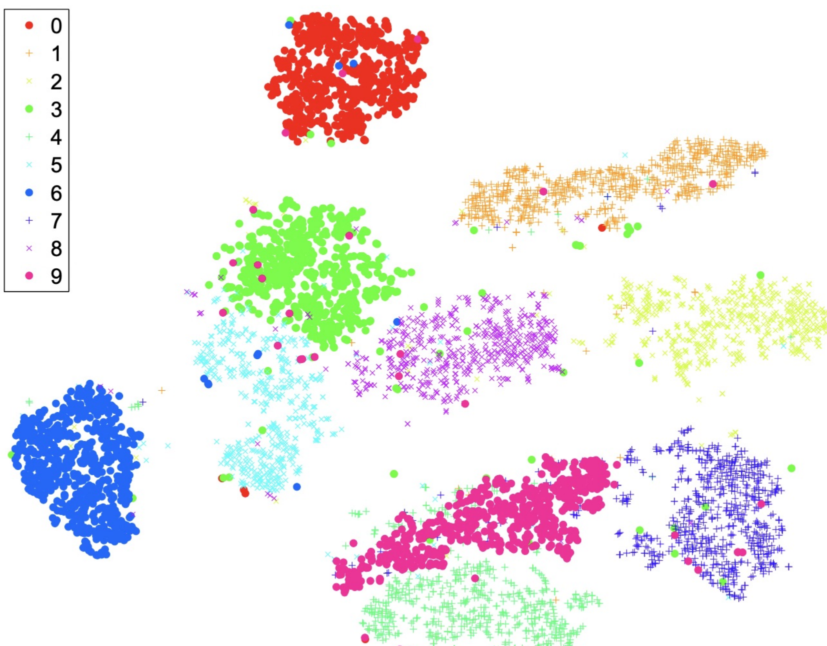


Visualization by PCA:





(a) Visualization by Isomap.



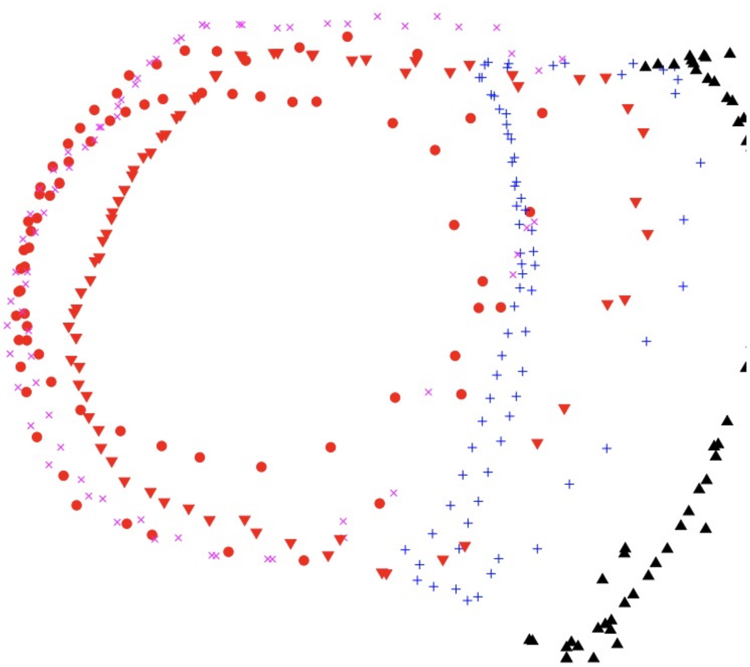
(a) Visualization by t-SNE.

[van der Maarten-Hilton 2008]

Example: COIL dataset = images of rotated objects:

(a) Example images of twenty categories

(b) Part example images of one category



(c) Visualization by Isomap.

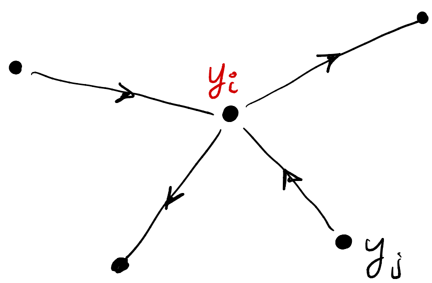
(a) Visualization by t-SNE.

- No theory so far.

- $D(P\|Q)$ is minimized by gradient descent.

$$\frac{\partial C}{\partial y_i} = 4 \sum_{j=1}^{n} (p_{ij} - q_{ij})\, q_{ij}\, (y_i - y_j)$$

$\Rightarrow$ y_i is <u>attracted</u> or <u>repelled</u> by each other point y_j

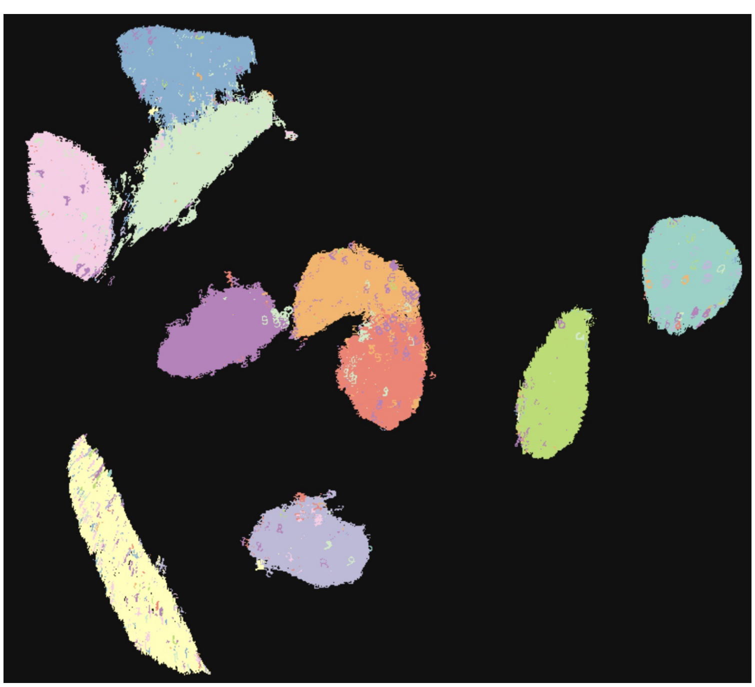


Springs $\Rightarrow$ forces.

- $\Rightarrow$ can replace optimization by an explicit attraction-repulsion rule. $\Downarrow$

9 **UMAP** [McInnes et al. 2018]

MNIST data (umap-explorer)



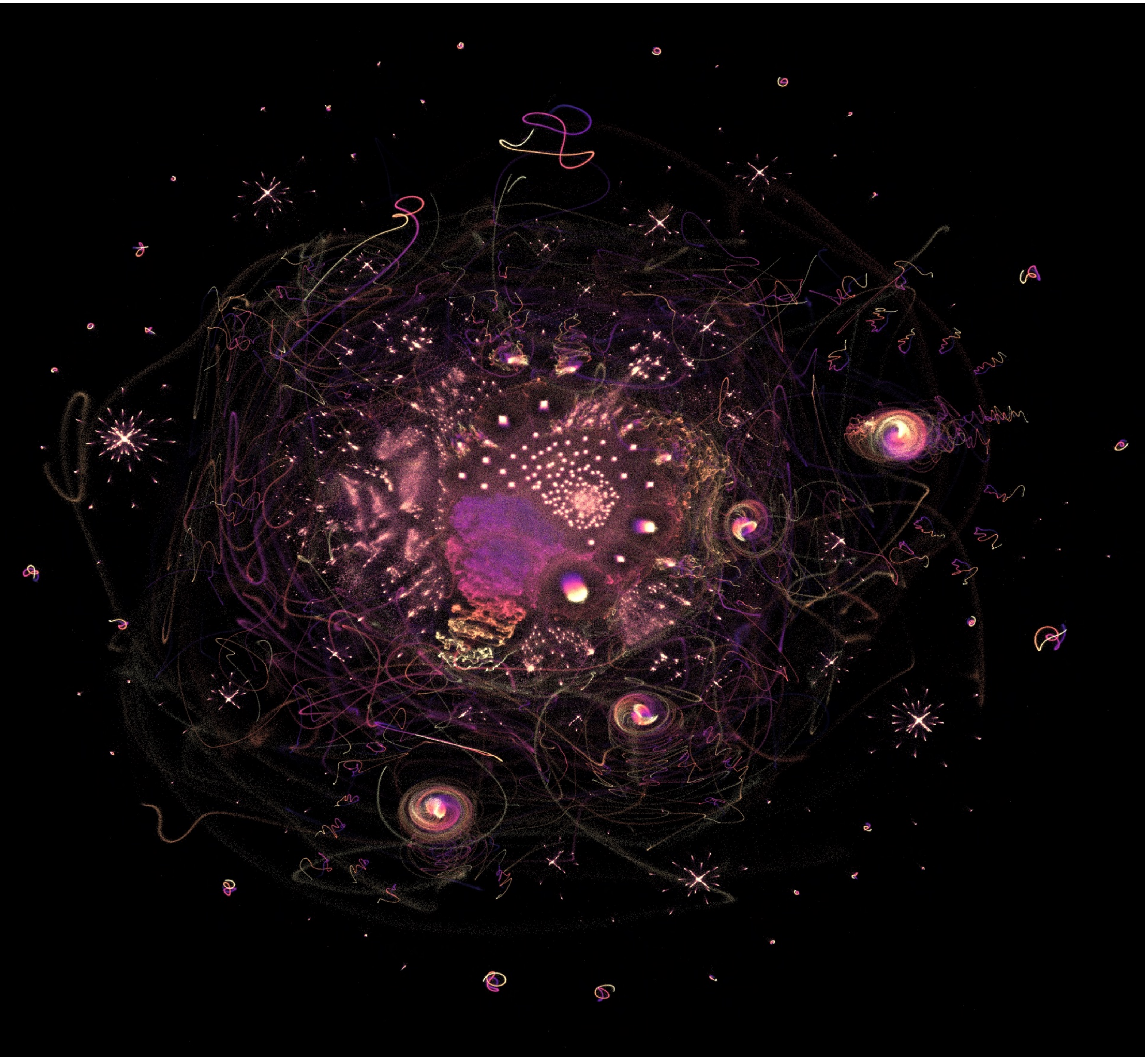
COIL data:

(10) What do numbers look like?

$n \in \mathbb{N} \longrightarrow$ Binary vector (prime factors)

$$\begin{array}{lcccccl} & 2 & 5 & 7 & 11 & 13, & \ldots \\ n = \underset{\substack{\| \\ 2\cdot 5\cdot 11}}{110} \longmapsto (& 1, & 1, & 0, & 1, & 0, & 0 \cdots) \end{array}$$

The first million integers visualized by UMAP [Williamson 2018]



(11) Fundamental questions:

- Is this all for real (galaxies etc.) or fake, a creation of the method?

  ⇓

- Theory?
- What is a good visualization? Is there "the best one"?
- What are clusters? How many? (e.g. types of cancer)