Geometric data analysis

Lecture 3/7 – Graphs

Jean Feydy
HeKA team, Inria Paris, Inserm, Université Paris-Cité

Thursday, 9am–12pm – 7 lectures

Faculté de médecine, Hôpital Cochin, rooms 2001 + 2005

Validation: project + quizz

Recap of the first two lectures

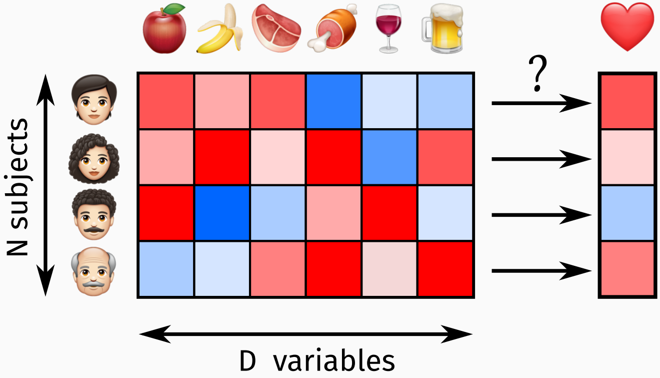
Lecture 1 – **Introduction**:

- AI = model + data.
- A good model is simple, accurate and **honest**.
- Understanding your model is key to **creativity**.

Lecture 2 – **Flat** vector spaces:

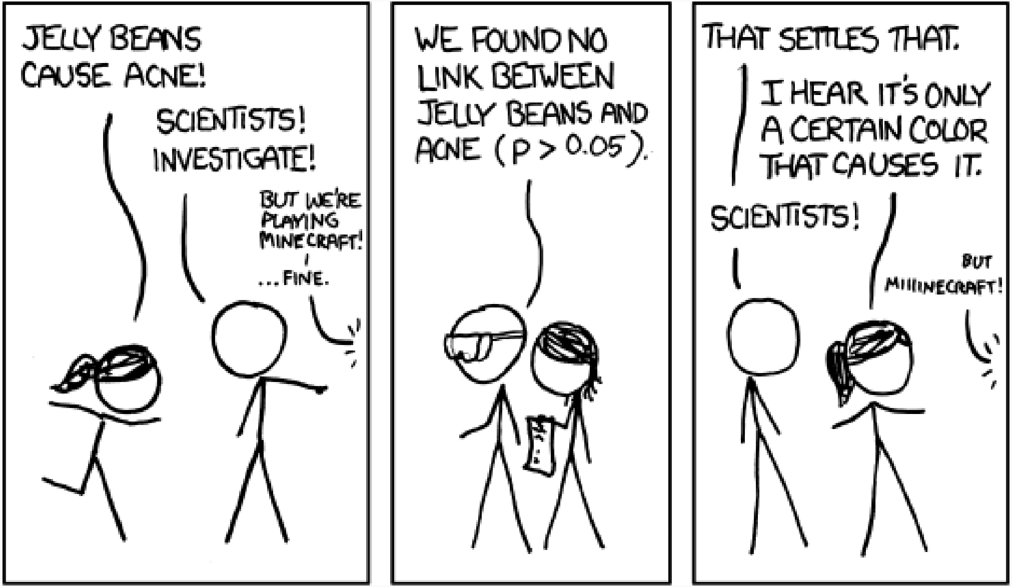
- Talk to domain experts ♡
- Best-case scenario: high-quality, informative features.
- Well-understood **baselines**: trees, K-NNs, linear and kernel regression.

What about the curse of dimensionality?

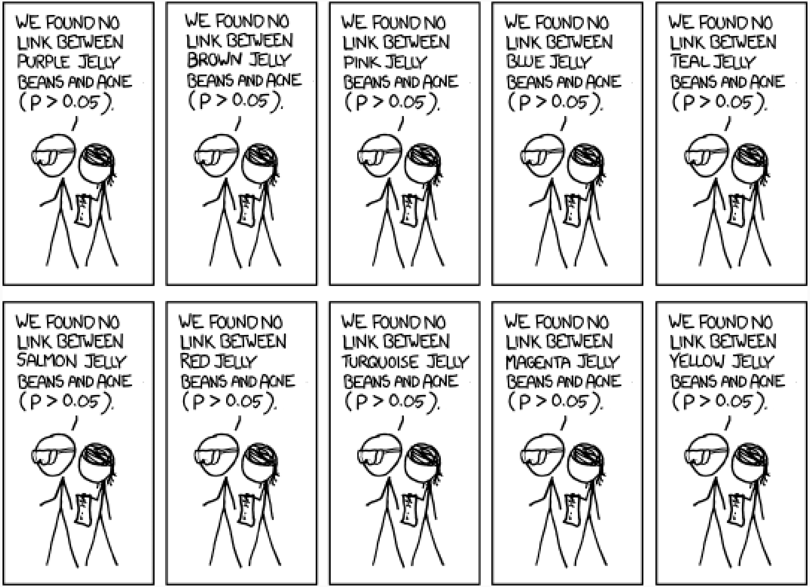


Remember: machine learning is about tables
that have **more columns than rows**.

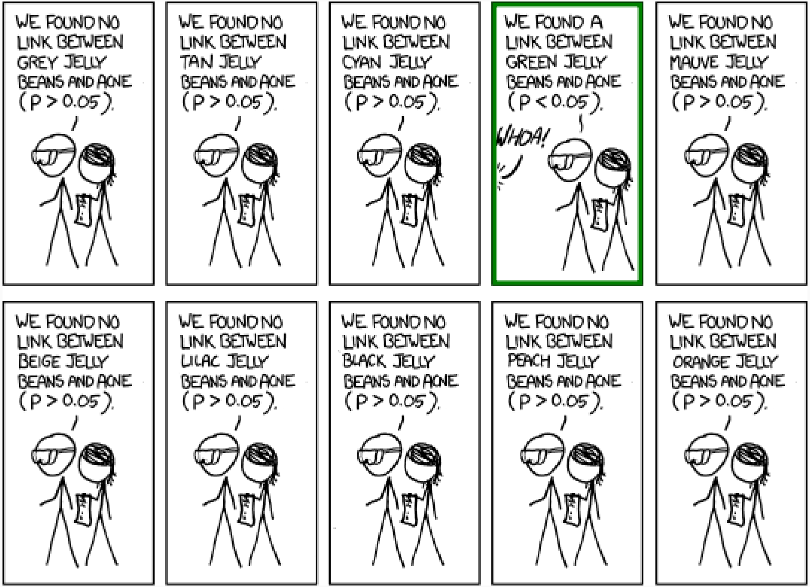
The statistical curse of dimensionality (XKCD 882)



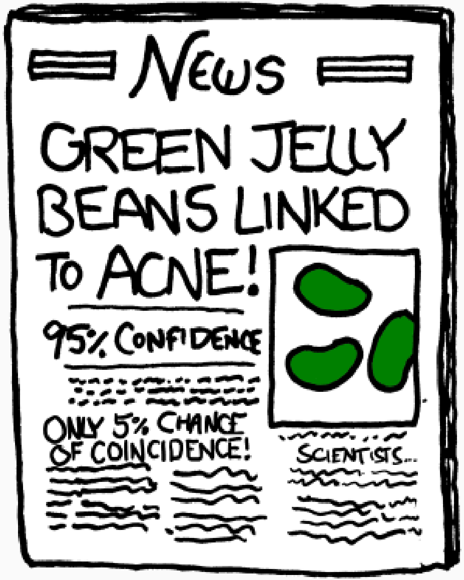
The statistical curse of dimensionality (XKCD 882)



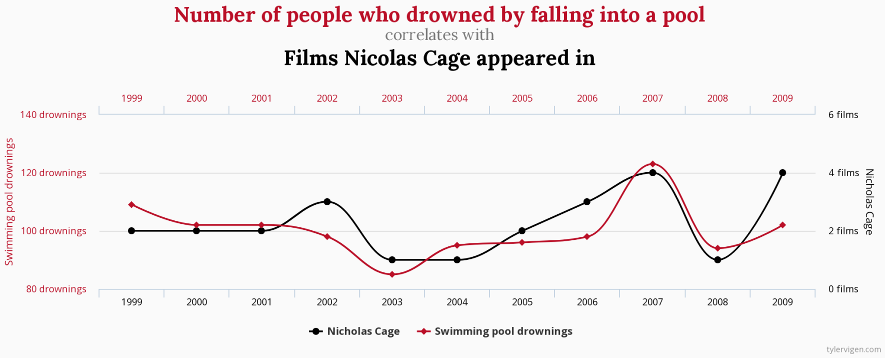
The statistical curse of dimensionality (XKCD 882)



The statistical curse of dimensionality (XKCD 882)



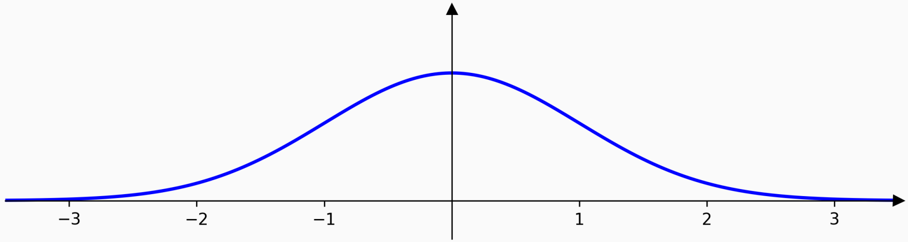
The statistical curse of dimensionality [Vig]



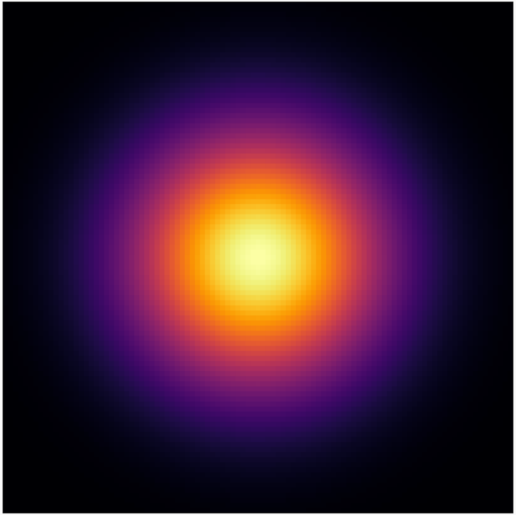
Two simple workarounds:

- **Sparsity**: trees, lasso…
- **Smoothness**: polynomial regression, kernels…

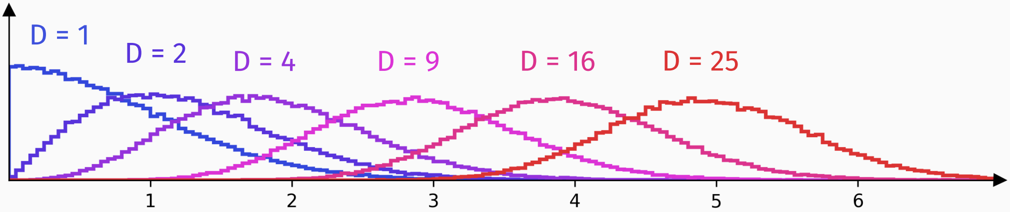
What does a Normal distribution look like... in dimension 1?



What does a Normal distribution look like… in dimension 2?



What does a Normal distribution look like... in higher dimension?



Histograms for 100,000 points x with D features of the Euclidean norm:

$$\|\mathrm{x}\| \;=\; \sqrt{\mathrm{x}[1]^2 + \cdots + \mathrm{x}[\mathrm{D}]^2}\;.$$

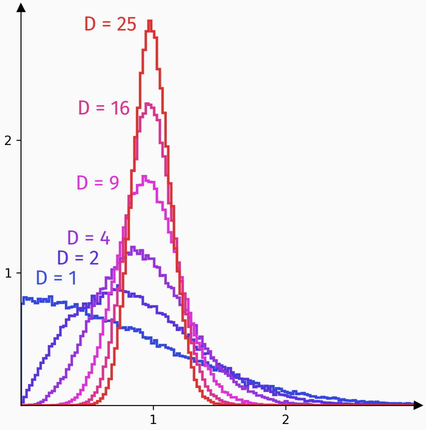
We recognize the **sum of D independent**, indentically distributed variables of **mean 1**.
Taking the square root, we get a random variable of mean $\sqrt{\mathrm{D}}$.

What does a Normal distribution look like... in higher dimension?

Histograms for 100,000 points x with D features of the scaled Euclidean norm:

$$\frac{1}{\sqrt{D}}\|x\| \;=\; \sqrt{\tfrac{1}{D}\left(x[1]^2 + \cdots + x[D]^2\right)}\,.$$

As predicted by the **central limit theorem**, $\frac{1}{\sqrt{D}}\|x\|$ concentrates around its mean value 1.

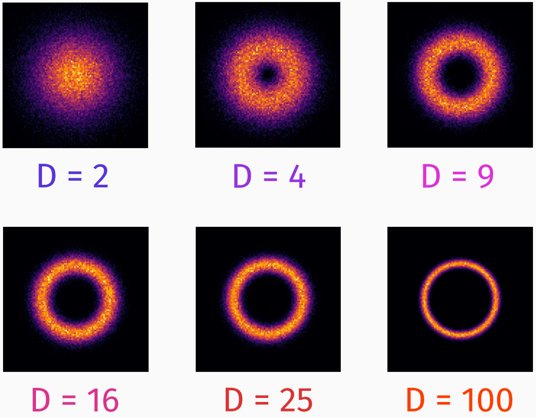


The “soap bubble” effect (Cf. Ferenc Huszár)

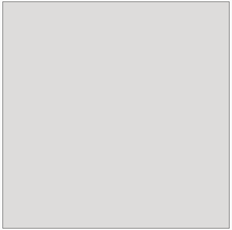
Histograms for 100,000 points x with D features of the rescaled planar projections:

$$\frac{\|x\|}{\sqrt{D} \cdot \sqrt{x[1]^2 + x[2]^2}} \left(x[1], x[2]\right) .$$

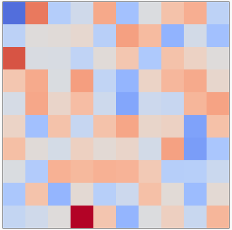
This provides a **faithful** visualization of a Normal Gaussian sample in dimension D.



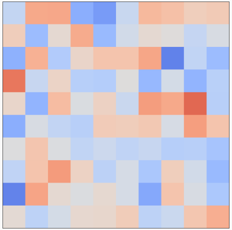
High-dimensional i.i.d. samples = white noise



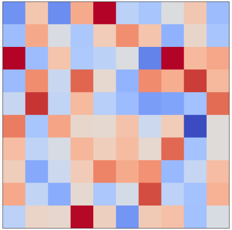
Mean value.



Sample x_1.



Sample x_2.



Difference $x_1 - x_2$.

Samples do **not** look like the average value of the distribution, and are all orthogonal to each other.

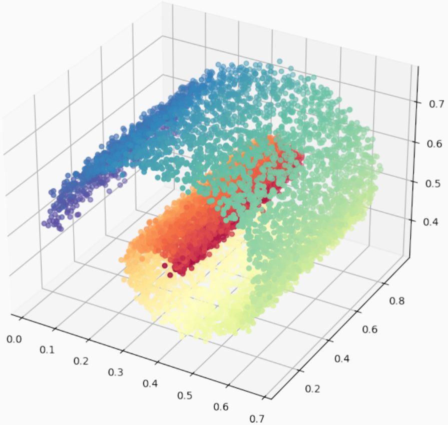
Geometric consequences of the curse of dimensionality

If we assume that our $D > 10$ features are **independent** and identically distributed:

- We require 10^D samples to enable basic statistics: histograms, density estimation…
- $\|x_i - x_j\|$ is **constant** up to a minor deviation.
- The distance matrix contains no useful information.
- All of **our intuitions break down**.

White noise is useless: garbage in, garbage out (XKCD 1838)

The manifold hypothesis



What matters for statistics is the **intrinsic** dimension “d” of the dataset, not the **extrinsic** dimension “D” of the feature space.

Overview of the class

Coming next:

- Lecture 3: **“Discrete”** geometries = graphs.
- Lecture 4: Deep learning on graphs and point clouds.
- Lecture 5: **“Continuous”** geometries = manifolds.
- Lecture 6: Spaces of probability distributions.
- Lecture 7: **Hardware** bottlenecks.

The aim of the class is to let you bridge the gap between
“discrete” and “continuous” **descriptions**
of the underlying problem structures.

Today’s lecture

1. **Why do we care about graphs?**

2. **Local descriptors and archetypes:**

   - Dimension.
   - Curvature.

3. **Global embeddings:**

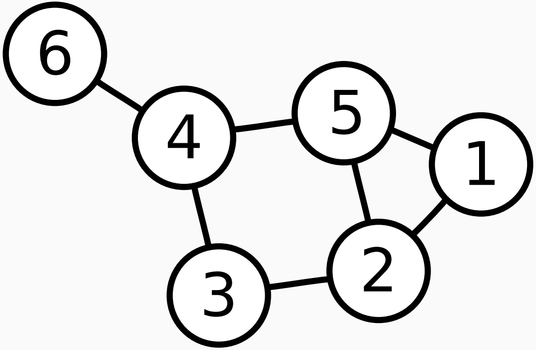
   - Lab session with UMAP.

Why do we care about graphs?

Yet another cake with our domain experts…

"We may not fully understand the columns of our table…
But we can certainly tell you that Patient A **is similar** to Patient B!"

Undirected graphs

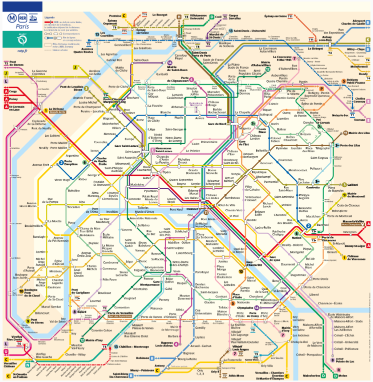


Algorithmic definition: a collection of vertices and edges.

Geometric perspective: a metric space that is defined **locally**…

and that we would like to understand **globally**!

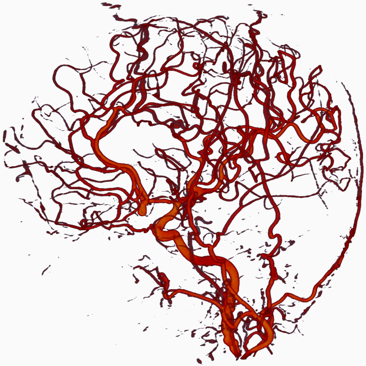
Networks and webs



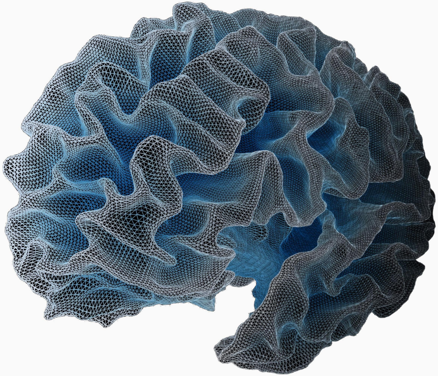
Transportation networks.

Communication and power lines.

3D curves and meshes [Fis12]



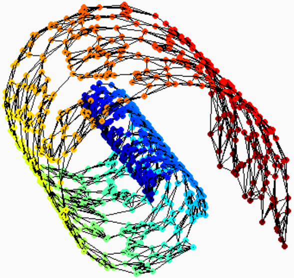
Vascular networks.



Anatomical surfaces.

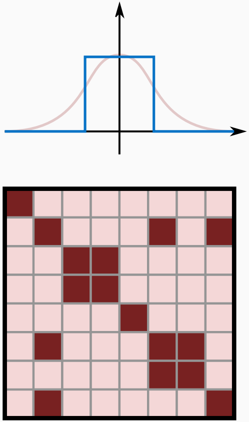
Intrinsic graph distances $\neq$ **Extrinsinc** Euclidean distances.

From high-dimensional samples to graphs: K-Nearest Neighbors [Pey11]

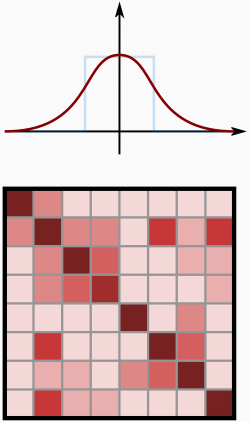


If your data has a low-dimensional structure,
this should be visible on its **neighborhood** structure.

From high-dimensional samples to graphs: kernel matrices

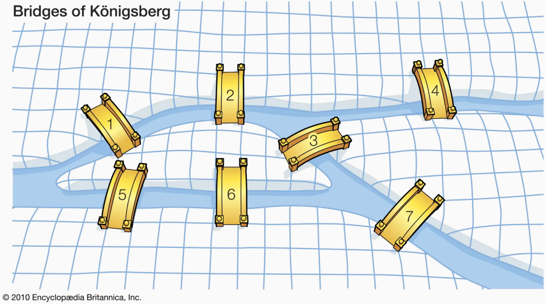


Ball connectivity matrix.

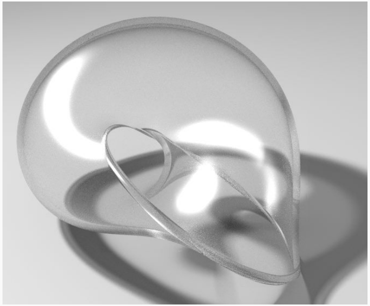


Gaussian kernel matrix.

Classical geometry [Bri, Che]



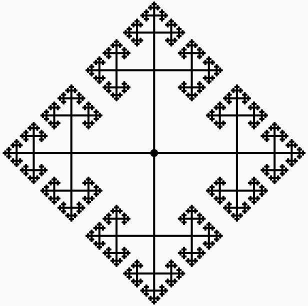
$\neq$



Discrete graph.

Continuous surface.

Modern geometry [Red]



$\simeq$

Tree.

Hyperbolic salad.

It is not the encoding that matters – but the geometry that is inside [SACO22]

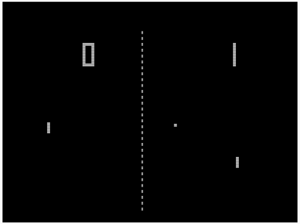
Mesh triangulations, sampling densities or point cloud representations should **not** distract ourselves from the underlying objects.

But how do we untangle such a web of vertices and edges? [Mat11]

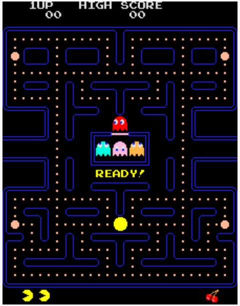
Going from **local** connectivity to **global** structure
is the main open challenge in geometry.

Dimension

Dimension = number of degrees of freedom



Pong is 1D.

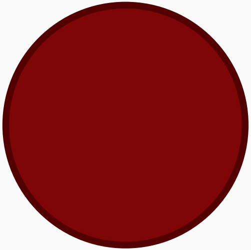


Pac-Man is 2D.

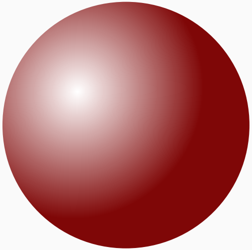
Minecraft is 3D.

What about geometric objects?

A segment is 1D.

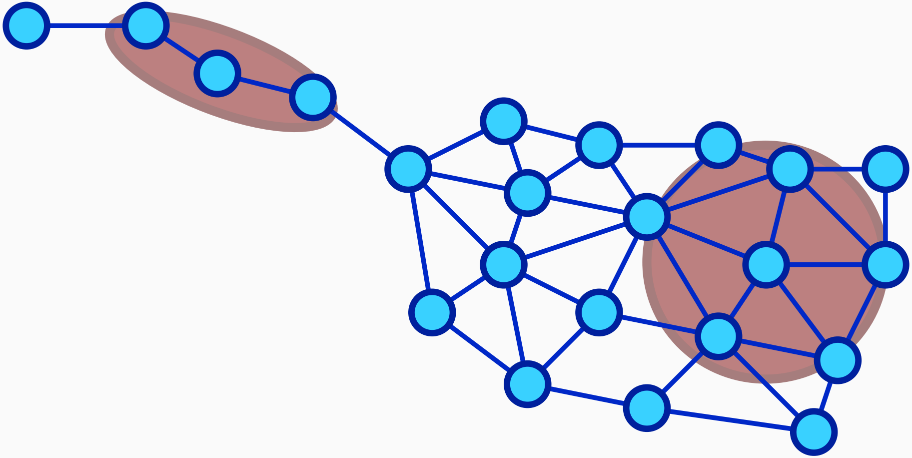


A disk is 2D.

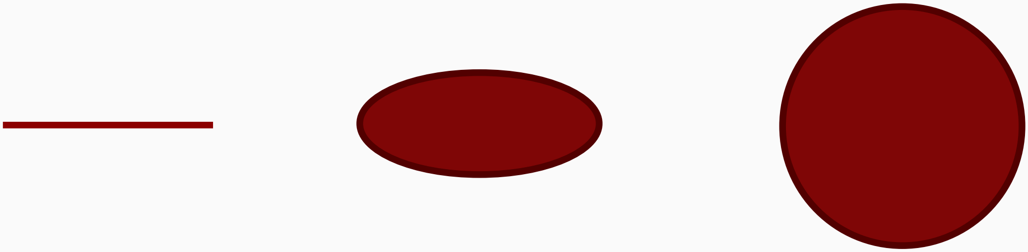


A ball is 3D.

Fitting an ellipsoid to a K-NN graph is easy using local PCA



What is the dimension of an ellipsoid?



d = 1

d = 1.5 ?

d = 2

A rule of thumb

Let $\lambda_1, \dots, \lambda_D$ denote the lengths of the principal axes.

$\lambda_1^2 > \dots > \lambda_D^2$ are the diagonal coefficients of the PCA.

We normalize them as $l_i = \lambda_i^2 / (\lambda_1^2 + \dots + \lambda_D^2)$.

Then, we may define the **local dimension** d as the smallest index such that

$$l_1 + \dots + l_d > 80\%.$$

Other conventions exist!

What about "pure" graphs?

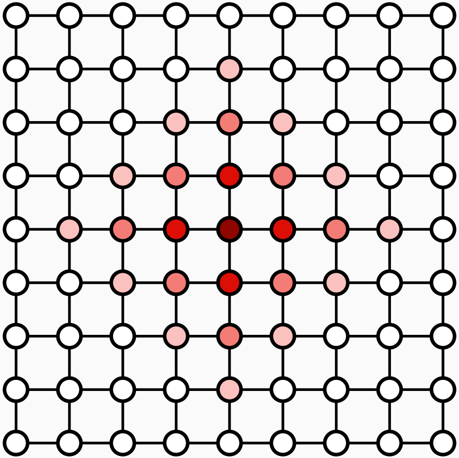
Hausdorff dimension: we pick d such that:

$$\mathrm{Vol}(B(\mathsf{x}, \mathsf{r})) \sim \mathsf{r}^{\mathsf{d}}.$$

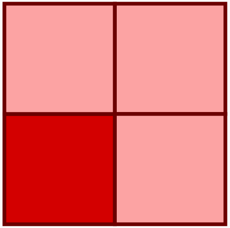
We estimate:

$$\mathsf{d} \simeq \frac{\log(\mathrm{Vol}(B(\mathsf{x}, \mathsf{r})))}{\log(\mathsf{r})}$$

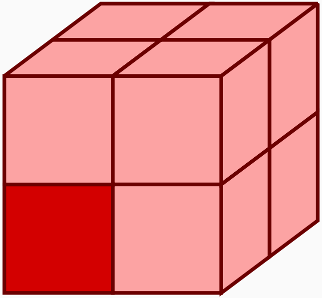
by a linear regression on $\mathsf{r} = 1, 2, 3, 4, 5, \ldots$



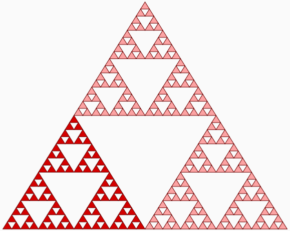
This definition works well for many objects, including fractals.



$$d = \frac{\log 4}{\log 2} = 2$$

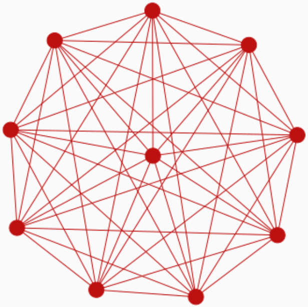
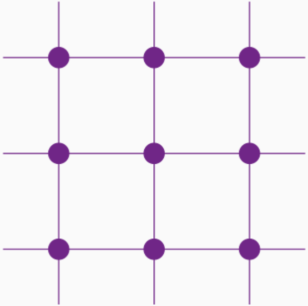
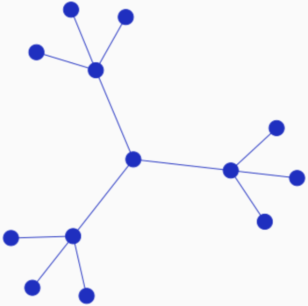


$$d = \frac{\log 8}{\log 2} = 3$$



$$d = \frac{\log 3}{\log 2}$$

Problem: $\mathrm{Vol}(B(\mathbf{x}, \mathbf{r}))$ is not always a polynomial function of r [TDGC+21]

Cliques: we fill the graph and **plateau** very quickly.

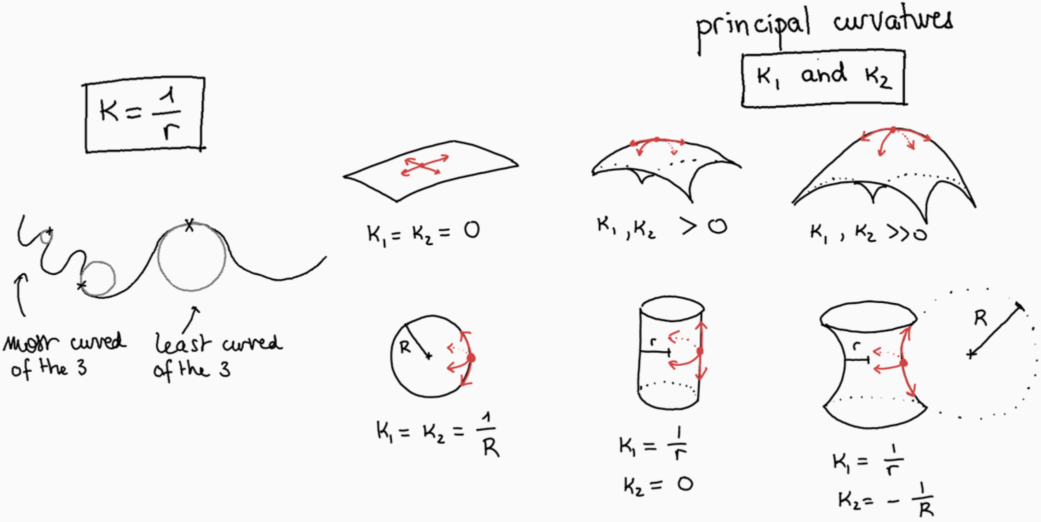
Grids: we retrieve a **polynomial**.

Trees: the volume of a ball grows **exponentially** fast.

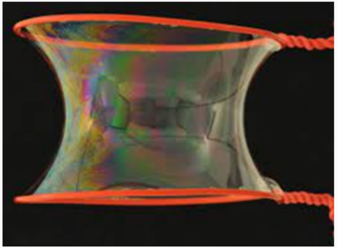
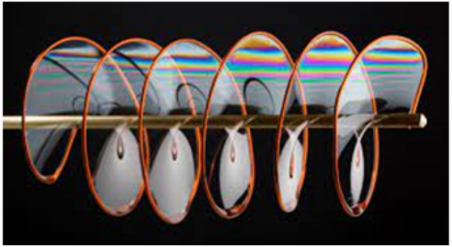
This reminds us of **classical examples** in continuous geometry: the sphere, the Euclidean plane and the Poincaré disk.

Curvature

Curvature of a 2D surface



Direct uses in physics and biology

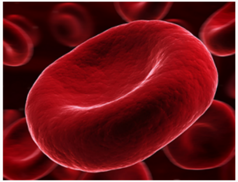
Soap bubbles minimize:

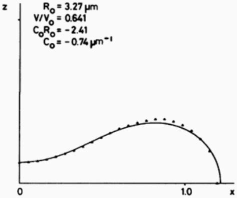
$$\mathrm{area}(\mathcal{S}) = \int_{\mathcal{S}} 1\, dA$$

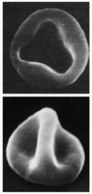
under constant volume, or with boundary conditions.

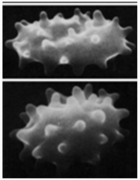
They correspond to minimal surfaces with $H = \kappa_1 + \kappa_2 = 0$ in cases 2 and 3.

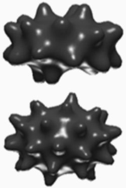
Direct uses in physics and biology









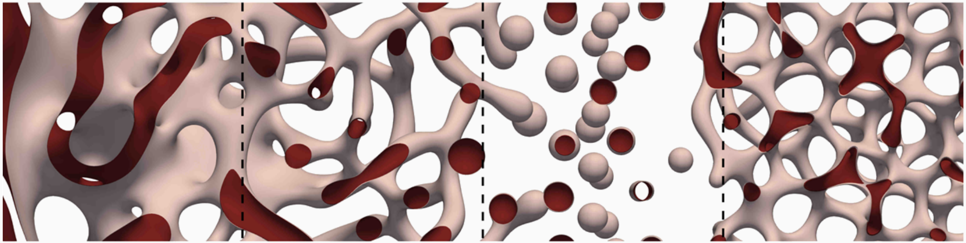


Red blood cells minimize:

$$\mathrm{Helfrich}(\mathcal{S}) = \int_{\mathcal{S}} (H - H_0)^2 \, dA$$

or a variant of this energy, under constant volume.

Direct uses in physics and biology [Son22]



$H^2 + \kappa_1^2 + \kappa_2^2$ | $(H-20)^2 + 5\,\kappa_2^2$ | $(H-25)^2 + (\kappa_1 - 12.5)^2 + (\kappa_2 - 12.5)^2$ | $(H-45)^2 + (\kappa_1 - 45)^2 + 10\,\kappa_2^2$

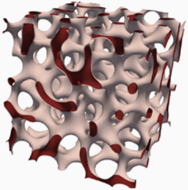
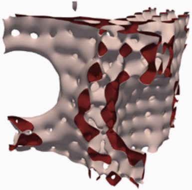
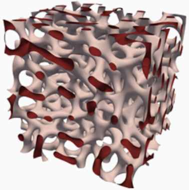
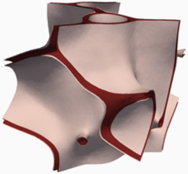
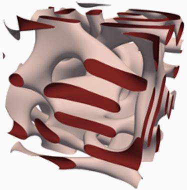
Curvatubes minimize:

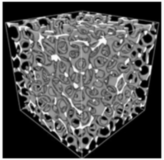
$$F(\mathcal{S}) = \int_{\mathcal{S}} p(\kappa_1, \kappa_2)\, dA$$

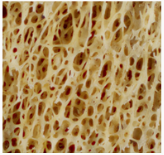
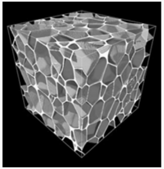
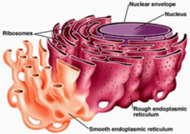
under constant volume, where:

$$p(\kappa_1, \kappa_2) = a_{2,0}\,\kappa_1^2 + a_{1,1}\,\kappa_1\kappa_2 + a_{0,2}\,\kappa_2^2 + a_{1,0}\,\kappa_1 + a_{0,1}\,\kappa_2 + a_{0,0}\,.$$

Curvature is a powerful descriptor



μCT image of open aluminium foam

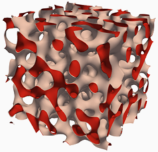
lamina cribrosa behind the eye

trabecular bone

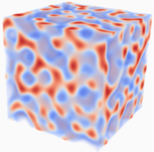
μCT image closed polymer foam

endoplasmic reticulum

Sidenote: your skillset goes way beyond deep learning research



$\longrightarrow$



$$\mathbf{F}(S) = \int_S p(\kappa_1, \kappa_2)\, dA$$

2D surface energy

$$\mathcal{E}_\epsilon(u) = \int_\Omega p(\kappa^\epsilon_{1,u}, \kappa^\epsilon_{2,u})\, \epsilon |\nabla u|^2\, dx$$

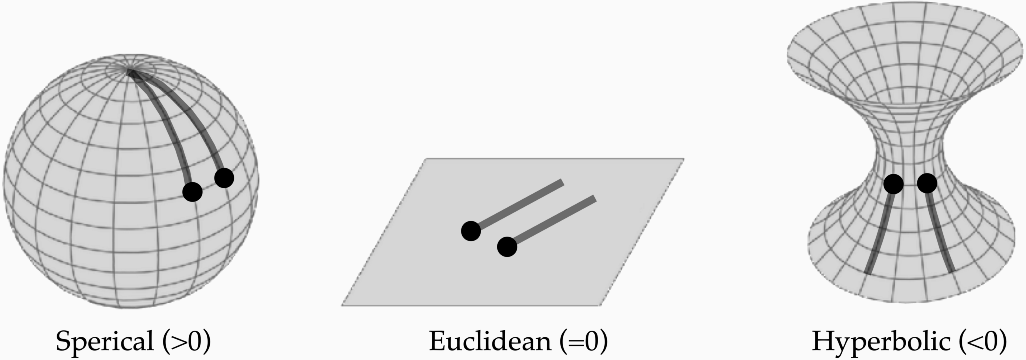
3D phase-field energy

An **inspiring** model:

- Surface energy $\rightarrow$ convolutional volumetric loss function (phase-field).
- Start with white noise (texture generation) and minimize with gradient descent.
- Implemented on GPU with PyTorch.

Combines maths + GPU computing + imaging data $\Longrightarrow$ Perfectly within your reach!

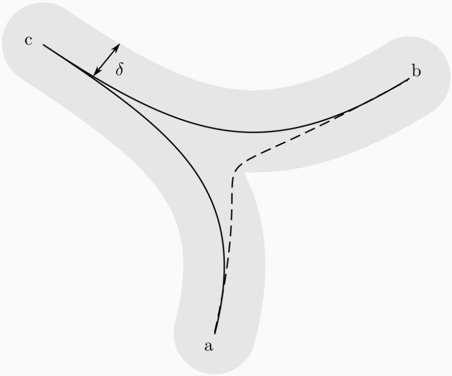
What about graphs? [TDGC+21]

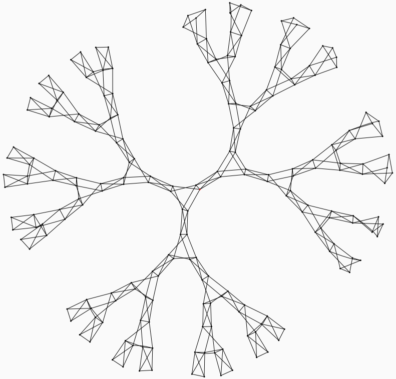


Theorem Egregium (Gauss, 1827):

- $H = \kappa_1 + \kappa_2$ is extrinsic $\rightarrow$ depends on the embedding.
- $K = \kappa_1 \cdot \kappa_2$ is intrinsic $\rightarrow$ can be defined on graphs.

Gromov's hyperbolicity [Gro87]

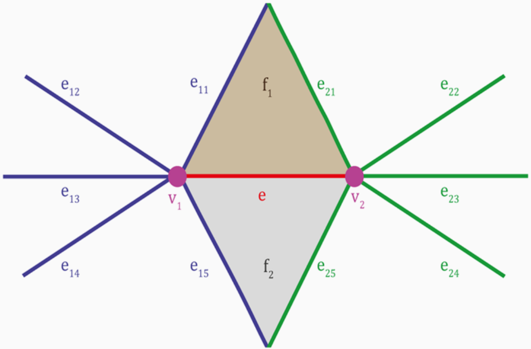




A graph is δ-hyperbolic if
all geodesic triangles are **thin**.

Global definition, suited to the study
of groups such as $\mathrm{SL}_2(\mathbb{Z})$.

Forman curvature [SMJ+16, SSG+18, TDGC+21]

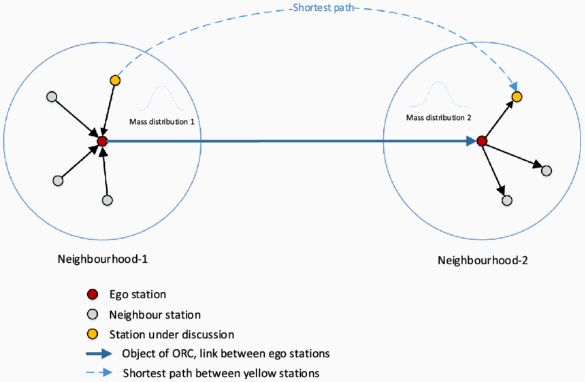


Forman curvature of an **edge** $\mathrm{i} \leftrightarrow \mathrm{j}$:

$$4 - \mathrm{degree}_{\mathrm{i}} - \mathrm{degree}_{\mathrm{j}} + 3 \cdot \mathrm{triangles}(\mathrm{i}, \mathrm{j}, \cdot\,)\ .$$

More precise but complex variations of this formula also exist.

Ollivier-Ricci curvature [WHY+22]

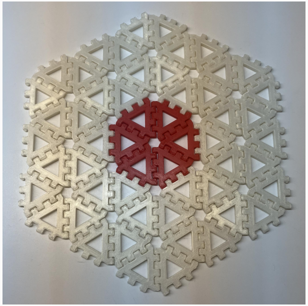


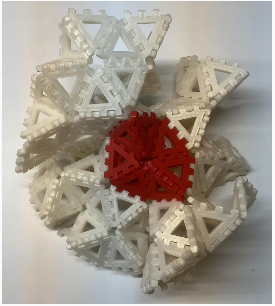
Ollivier-Ricci curvature of an **edge** $i \leftrightarrow j$:

is the optimal transport distance between $\mathcal{N}(x_i, 1)$ and $\mathcal{N}(x_j, 1)$

larger than the distance between x_i and x_j?

Toy models





5 triangles per vertex:
positive curvature.

6 triangles per vertex:
flat curvature.

7 triangles per vertex:
negative curvature.

$\Longrightarrow$ Basic intuition that guides current research in the field.

References

References i

Encyclopædia Britannica.

Bridges of königsberg.

https://www.britannica.com/topic/leisure/media/1/321794/68671.

Albert Chern.

Sudanese möbius band.

https://cseweb.ucsd.edu/ alchern/gallery/.

References ii

Bruce Fischl.

Freesurfer.

Neuroimage, 62(2):774–781, 2012.

Mikhael Gromov.

Hyperbolic groups.

In *Essays in group theory*, pages 75–263. Springer, 1987.

Matsuyuki.

Spaghetti picture.

https://commons.wikimedia.org/wiki/File:Aglio_e_olio.jpg, 2011.

CC BY-SA 2.0.

Gabriel Peyré.

The numerical tours of signal processing-advanced computational signal and image processing.

IEEE Computing in Science and Engineering, 13(4):94–97, 2011.

References iv

Allyson Redhunt.

Tessellations of the hyperbolic plane and m.c. escher.

https://web.colby.edu/thegeometricviewpoint/author/aredhunt/.

Nicholas Sharp, Souhaib Attaiki, Keenan Crane, and Maks Ovsjanikov.

DiffusionNet: Discretization agnostic learning on surfaces.

ACM Trans. Graph., 01(1), 2022.

RP Sreejith, Karthikeyan Mohanraj, Jürgen Jost, Emil Saucan, and Areejit Samal.
Forman curvature for complex networks.
Journal of Statistical Mechanics: Theory and Experiment, 2016(6):063206, 2016.

Anna Song.
Generation of tubular and membranous shape textures with curvature functionals.
Journal of Mathematical Imaging and Vision, 64(1):17–40, 2022.

Areejit Samal, RP Sreejith, Jiao Gu, Shiping Liu, Emil Saucan, and Jürgen Jost.

Comparative analysis of two discretizations of Ricci curvature for complex networks.

Scientific reports, 8(1):1–16, 2018.

Jake Topping, Francesco Di Giovanni, Benjamin Paul Chamberlain, Xiaowen Dong, and Michael M Bronstein.

Understanding over-squashing and bottlenecks on graphs via curvature.

arXiv preprint arXiv:2111.14522, 2021.

Tyler Vigen.

Spurious correlations.

https://www.tylervigen.com/spurious-correlations.

Yaoli Wang, Zhou Huang, Ganmin Yin, Haifeng Li, Liu Yang, Yuelong Su, Yu Liu, and Xv Shan.

Applying Ollivier-Ricci curvature to indicate the mismatch of travel demand and supply in urban transit network.

International Journal of Applied Earth Observation and Geoinformation, 106:102666, 2022.