MPRI – Computation Geometry and Topology

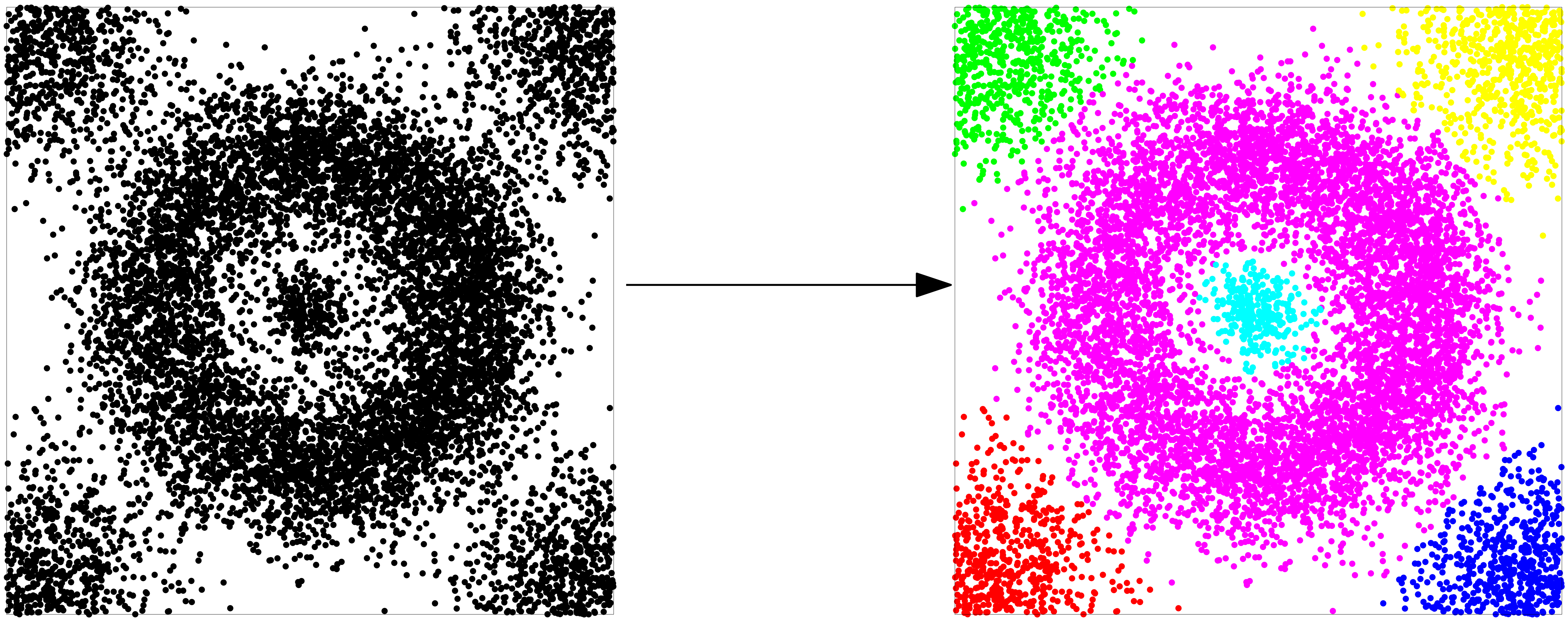
Clustering

Steve Oudot

(steve.oudot@inria.fr)

Cluster Analysis

Input: a finite set of observations:
- point cloud with coordinates
- distance / (dis-)similarity matrix



Task:

partition the data points into a collection of *relevant* subsets called clusters

A Wealth of Approaches

Variational

- k-means / k-medoid
- EM
- CLARA

spectral k-means

Spectral

- Normalized Cut
- Multiway Cut

Hierarchical divisive/agglomerative

- single-linkage
- BIRCH

Density thresholding

- DBSCAN
- OPTICS

Mode seeking

- Mean/Medoid/Quick Shift
- graph-based hill climbing

Valley seeking

- [JBD'79]
- NDDs [ZZZL'07]

A Wealth of Approaches

Variational

- k-means / k-medoid
- EM
- CLARA

spectral k-means

Spectral

- Normalized Cut
- Multiway Cut

Hierarchical divisive/agglomerative

- single-linkage
- BIRCH

Density thresholding

- DBSCAN
- OPTICS

Mode seeking

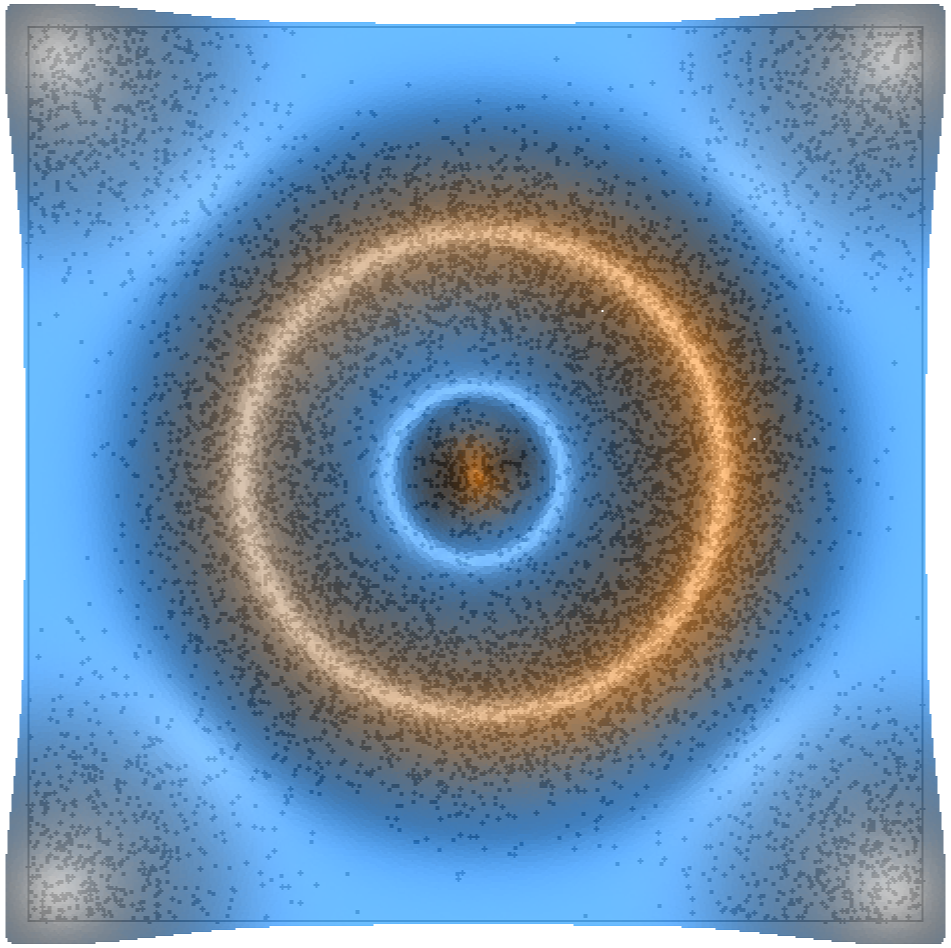
- Mean/Medoid/Quick Shift
- graph-based hill climbing

Valley seeking

- [JBD'79]
- NDDs [ZZZL'07]

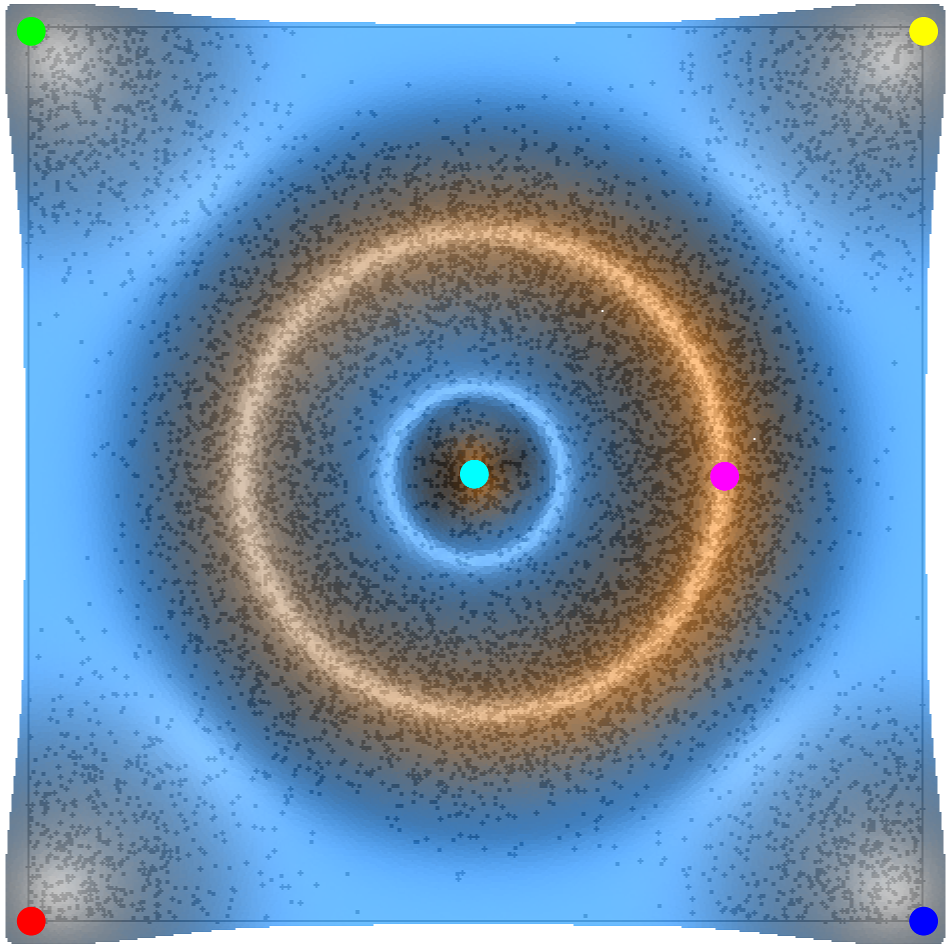
Mode-Seeking Paradigm

- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density



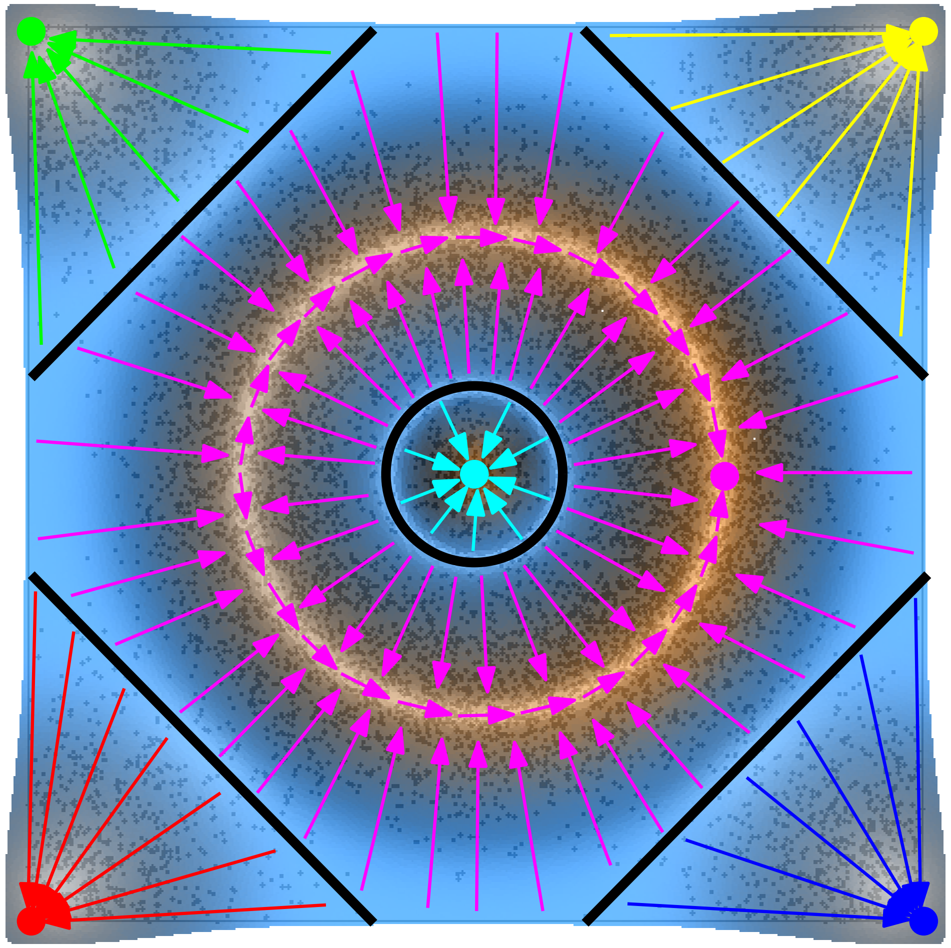
Mode-Seeking Paradigm

- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density



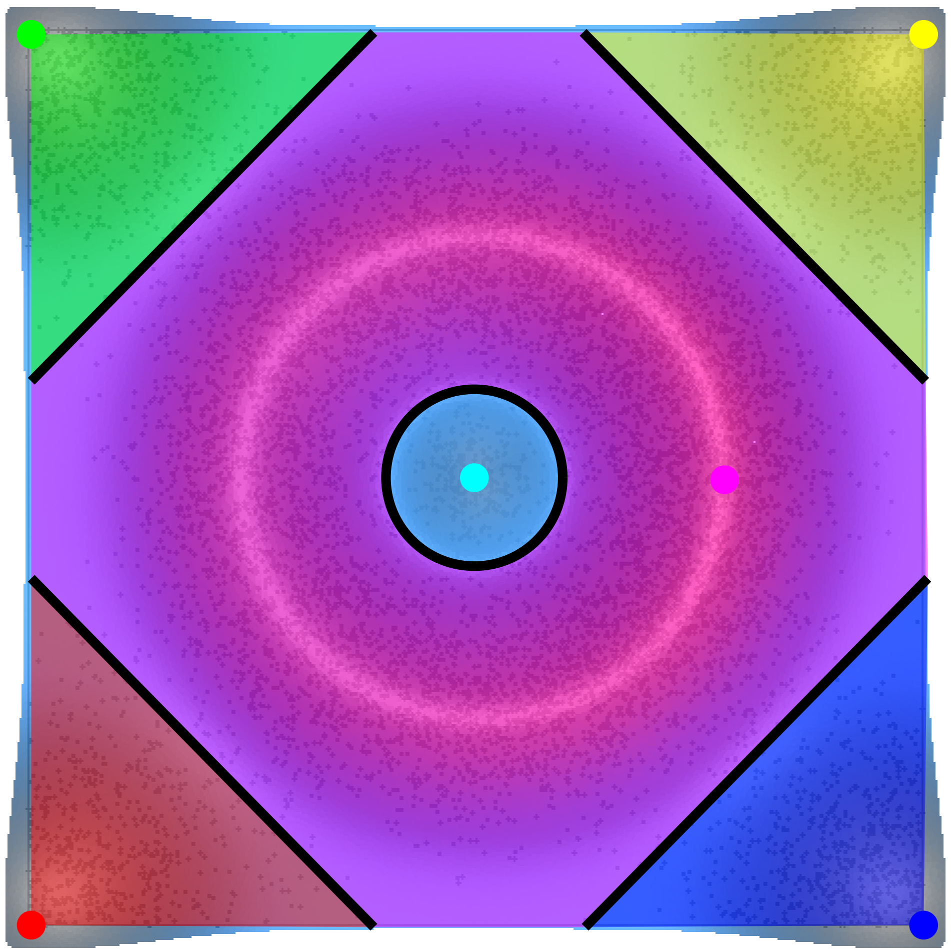
Mode-Seeking Paradigm

- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density



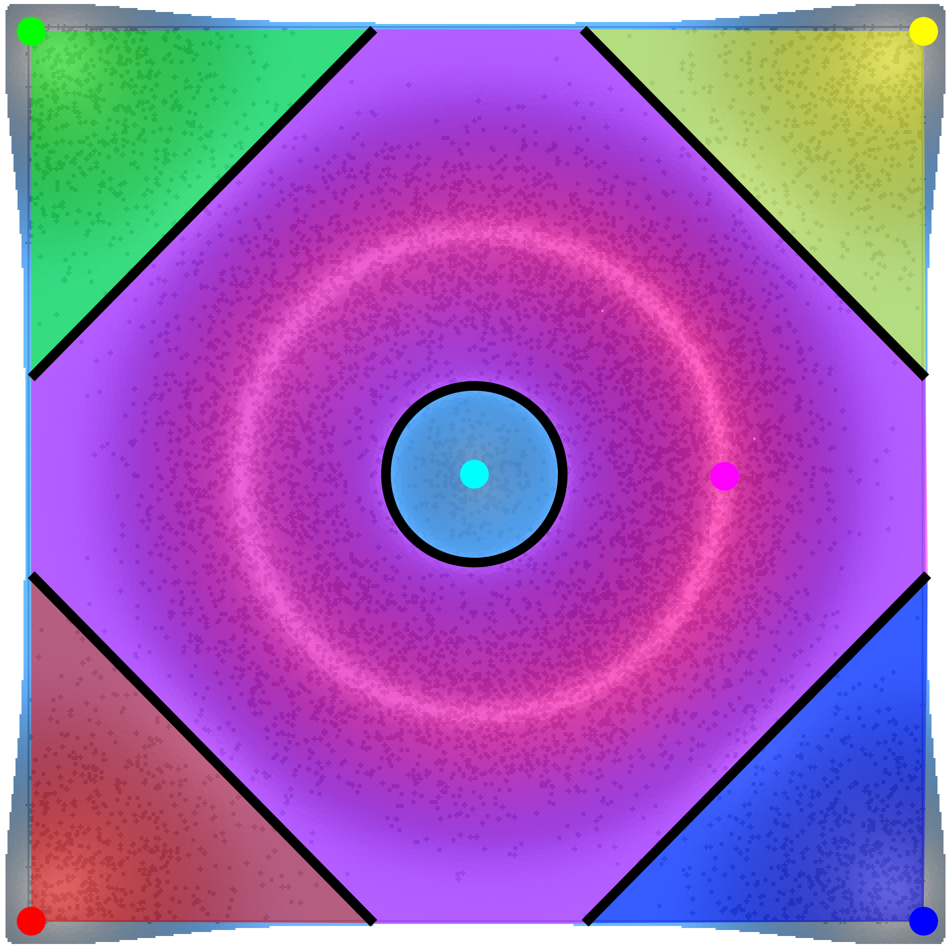
Mode-Seeking Paradigm

- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density

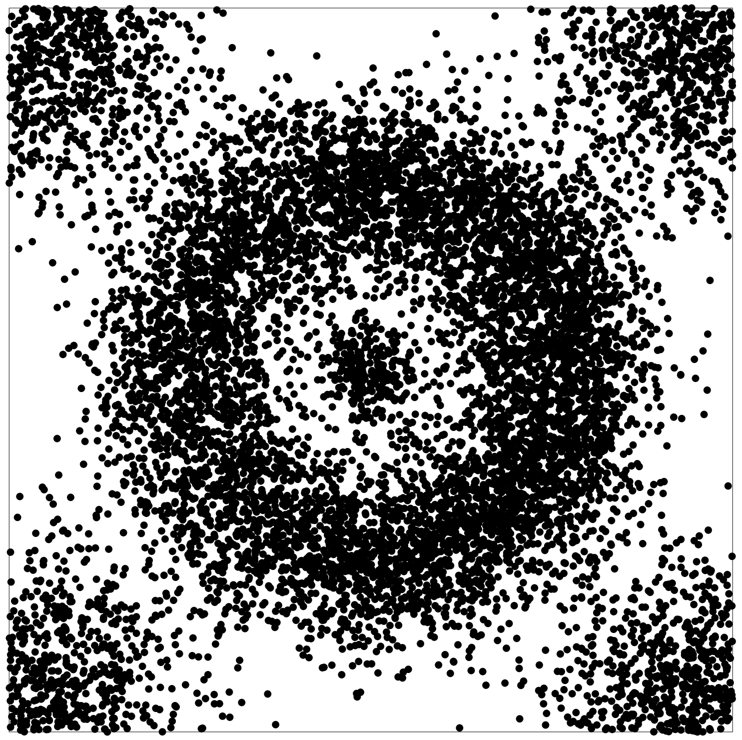


Mode-Seeking Paradigm

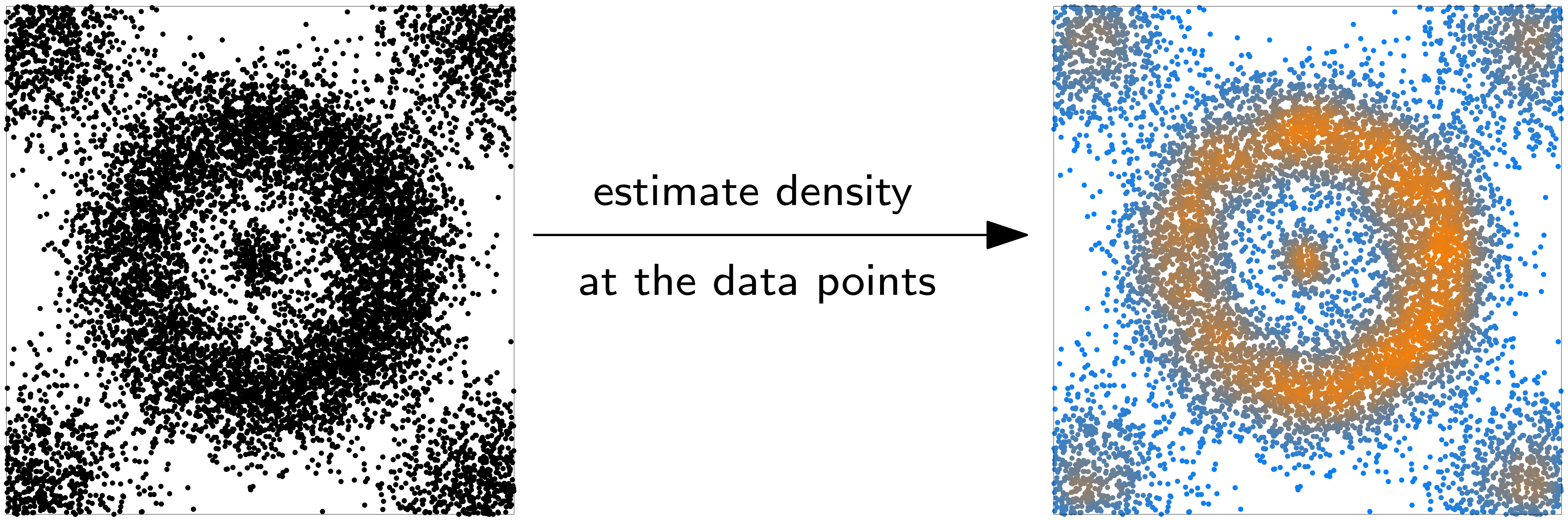
- Assume the data points are sampled from some unknown probability distribution
- Partition the data according to the basins of attraction of the peaks of the density



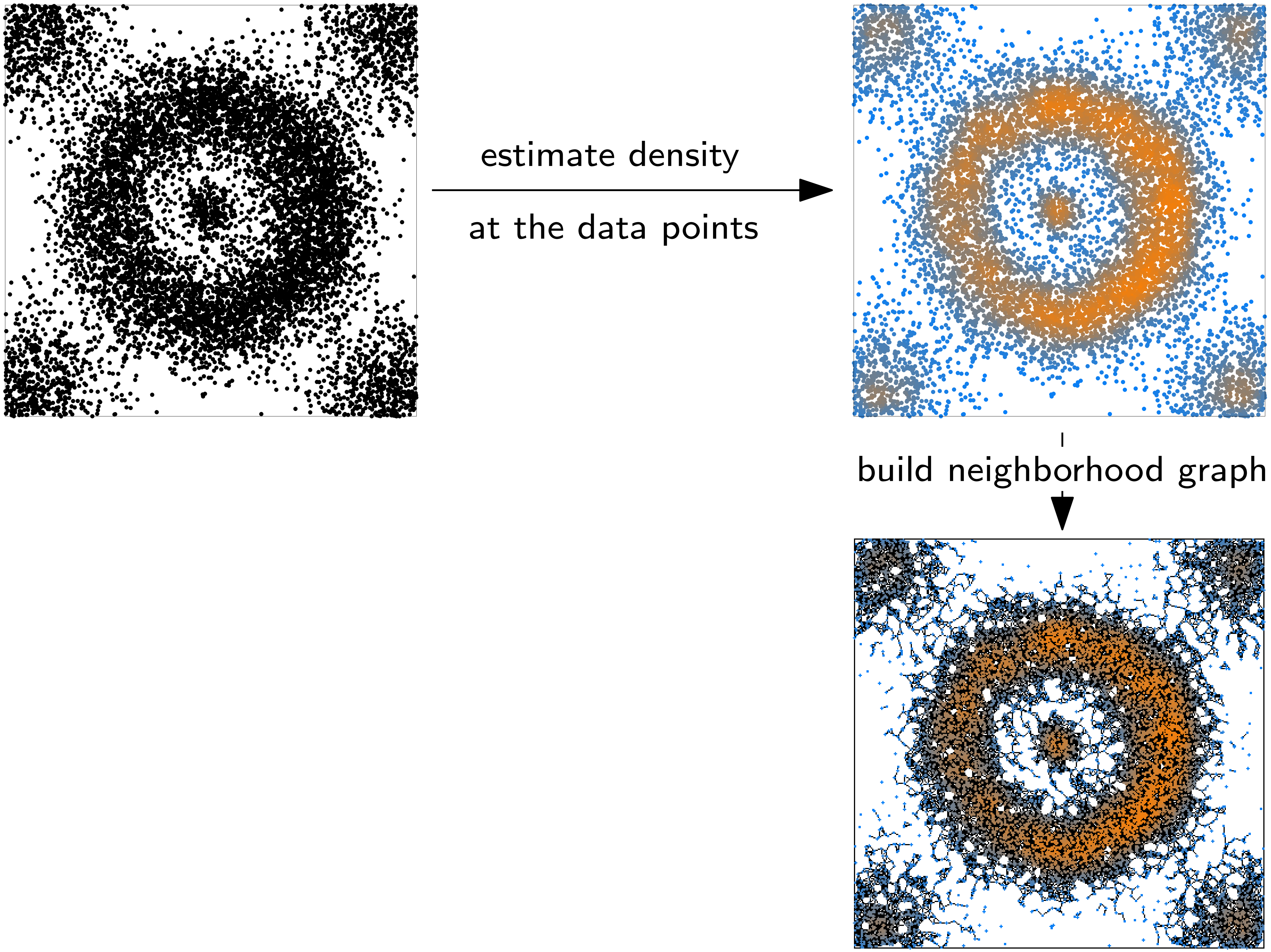
[Koontz, Narendra, Fukunaga'76] in a Nutshell



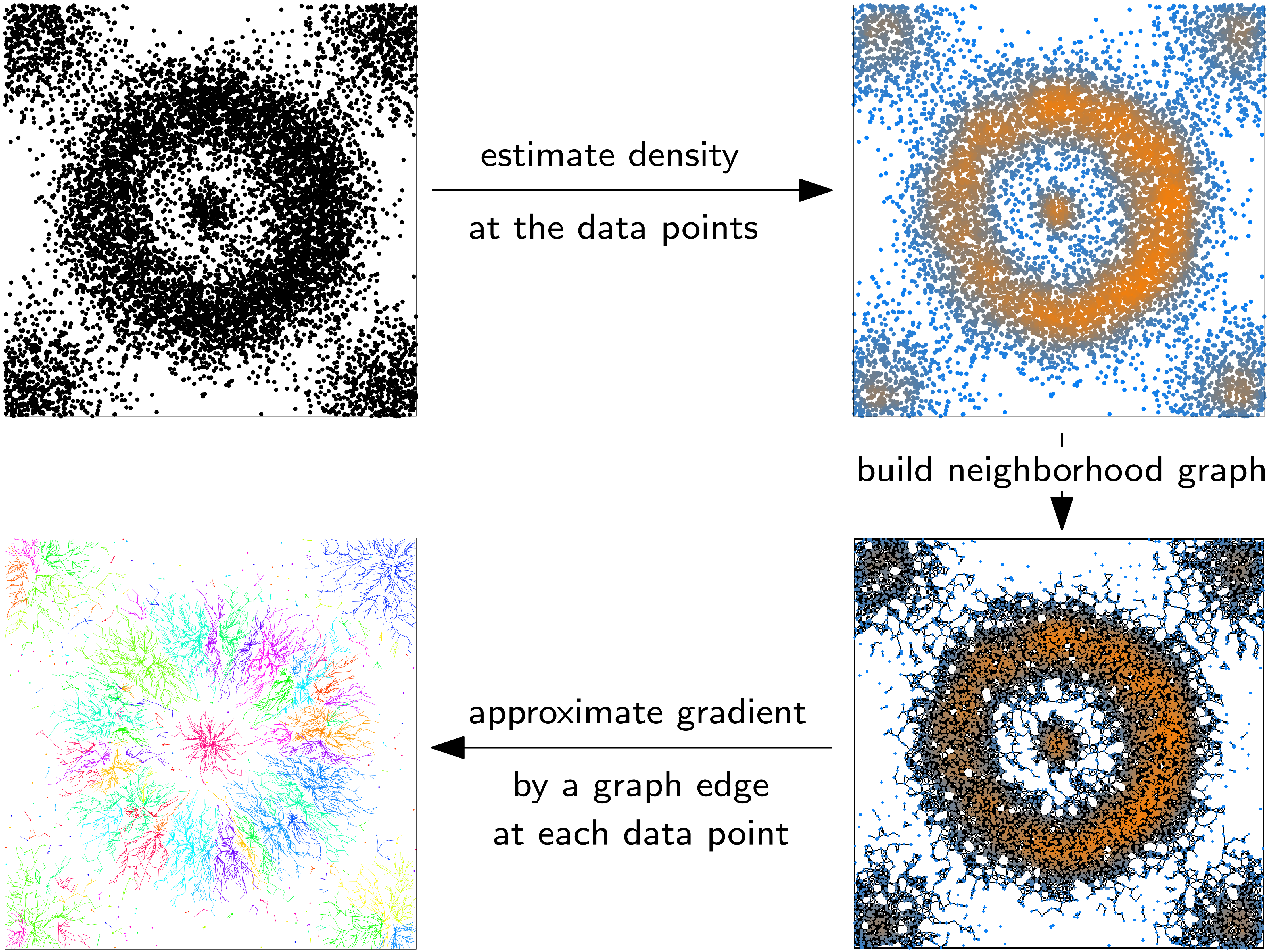
[Koontz, Narendra, Fukunaga'76] in a Nutshell



[Koontz, Narendra, Fukunaga'76] in a Nutshell

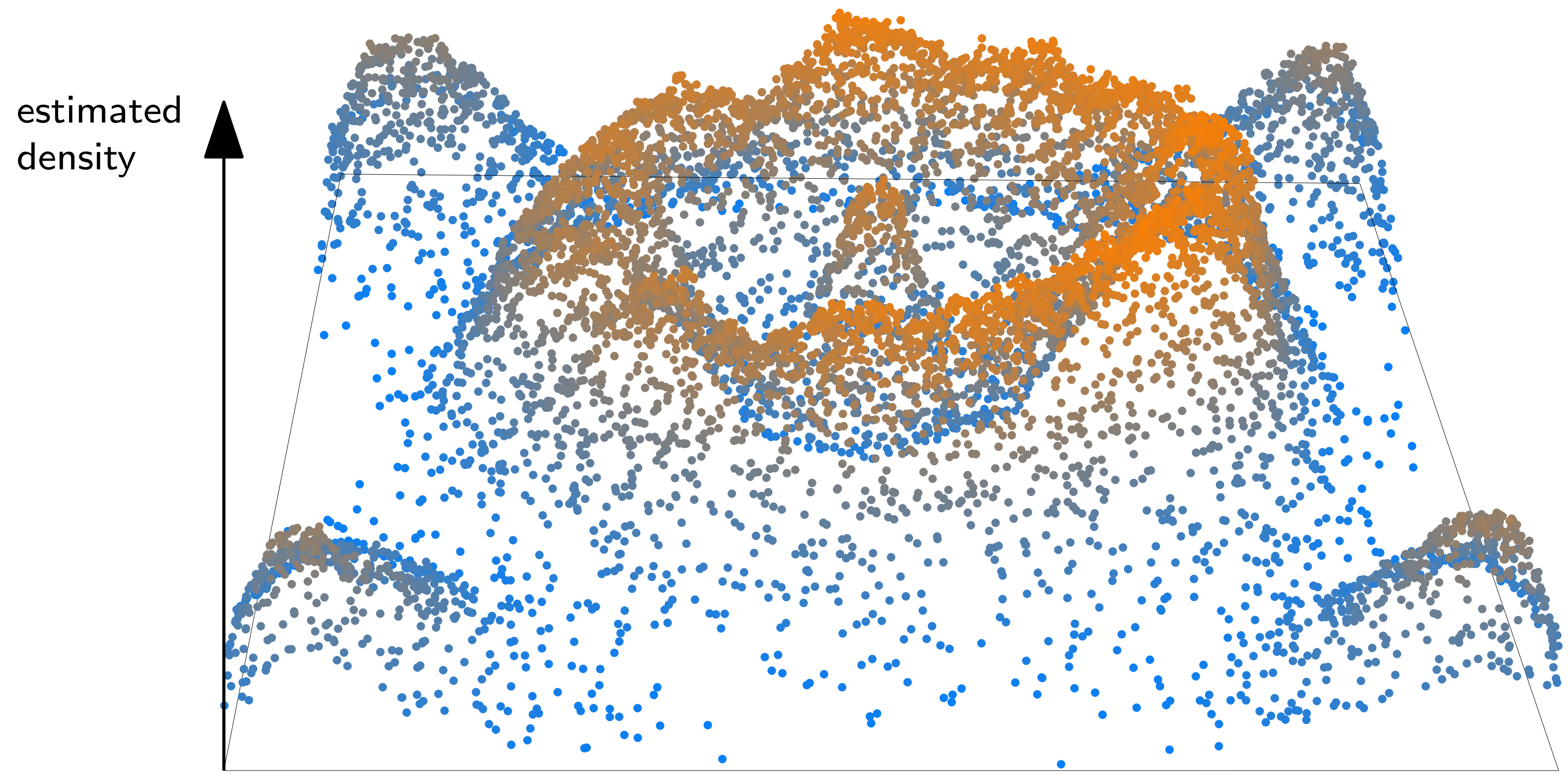


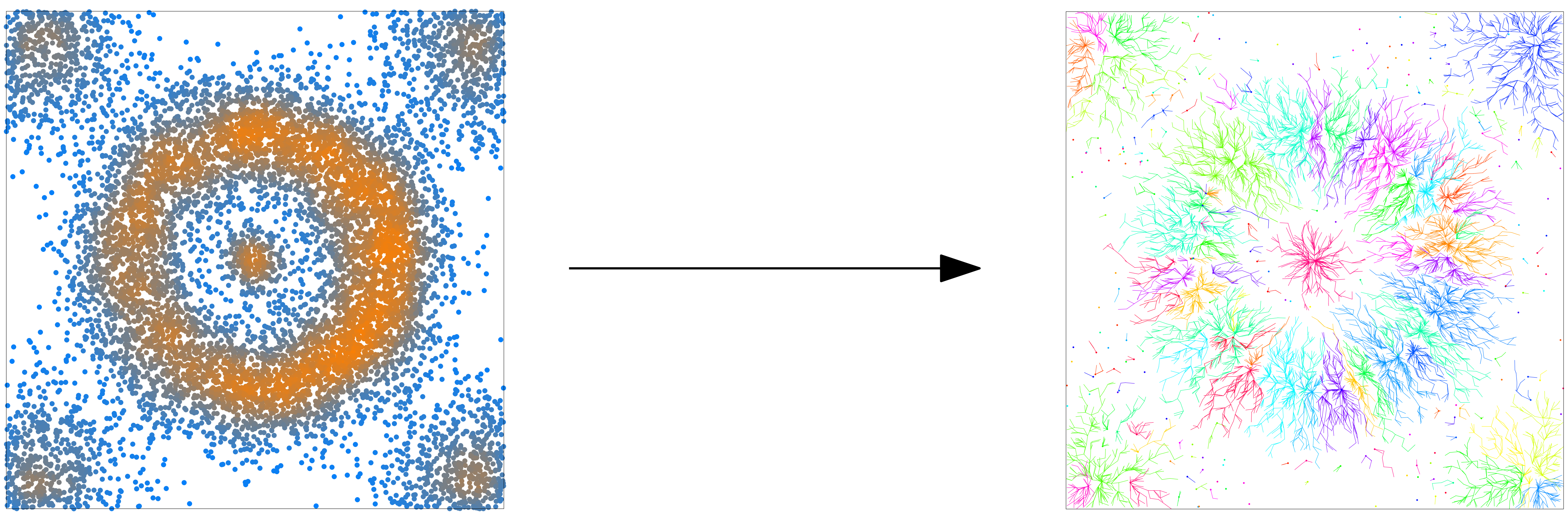
[Koontz, Narendra, Fukunaga'76] in a Nutshell



Why things are likely to go ill

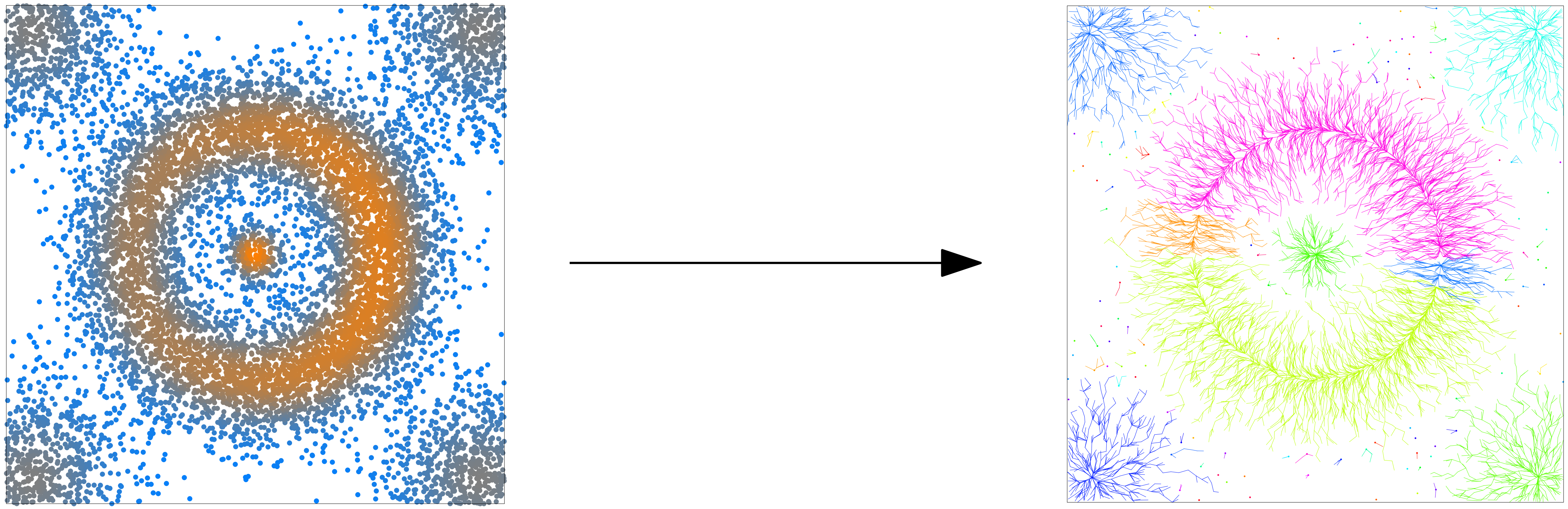
- Noisy estimator





Why things are likely to go ill

- Noisy estimator
- Neighborhood graph



Why things are likely to go ill

- Noisy estimator
- Neighborhood graph

Solutions:

1. **Be proactive**: act on approximate gradient flow (**Mean-Shift** [CM'02])

    $\rightarrow$ use kernel density estimator, with smoothing window parameter

    $\rightarrow$ work in ambient space to circumvent neighborhood graph issue

Why things are likely to go ill

- Noisy estimator
- Neighborhood graph

Solutions:

1. **Be proactive**: act on approximate gradient flow (**Mean-Shift** [CM'02])

   $\rightarrow$ use kernel density estimator, with smoothing window parameter

   $\rightarrow$ work in ambient space to circumvent neighborhood graph issue

2. **Be reactive**: merge clusters after clustering (**ToMATo** [CGOS'13])

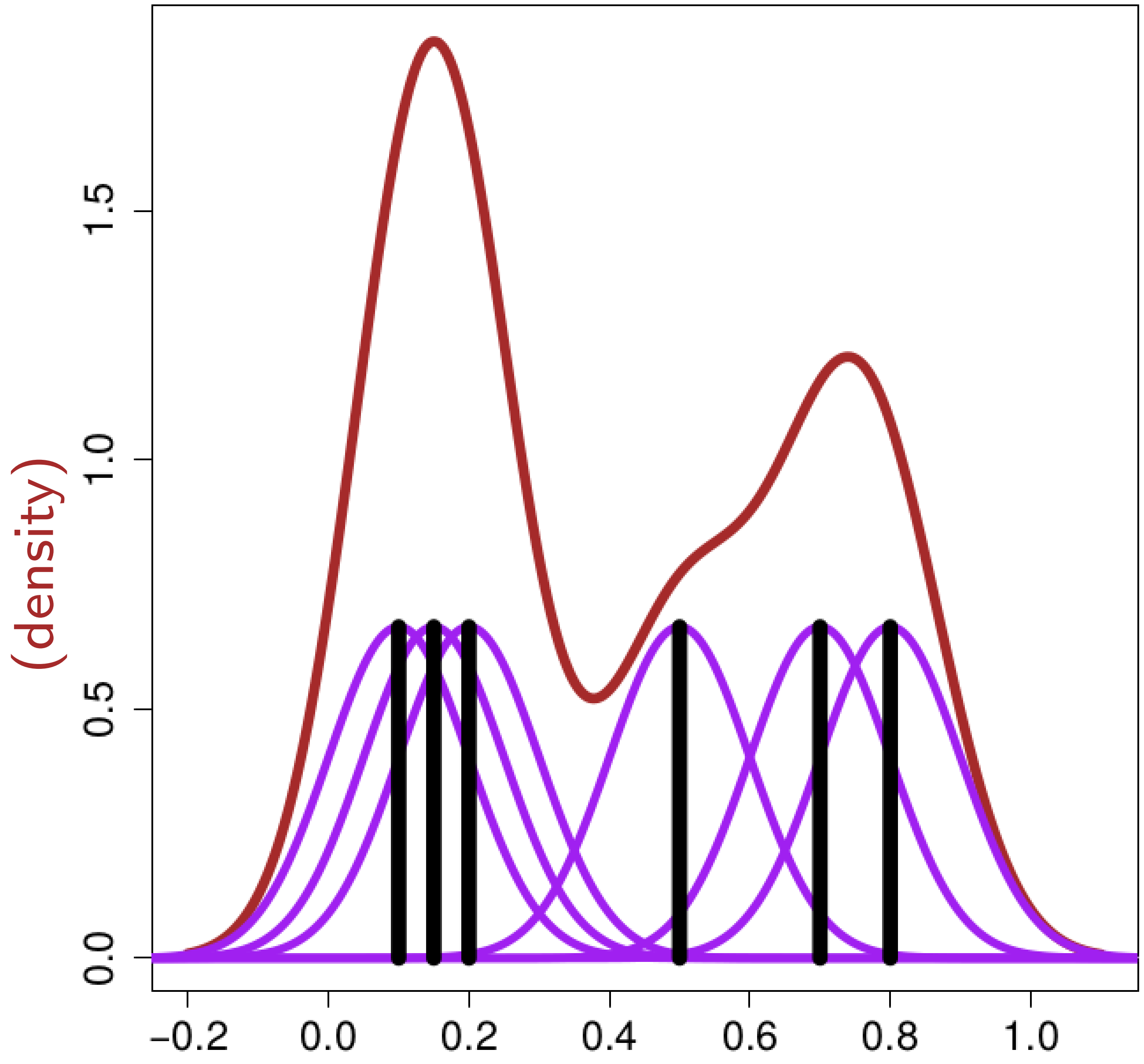
   $\rightarrow$ use topological persistence to guide a single-pass merging step

   $\rightarrow$ work in neighborhood graph to minimize prior knowledge

1. Mean-Shift

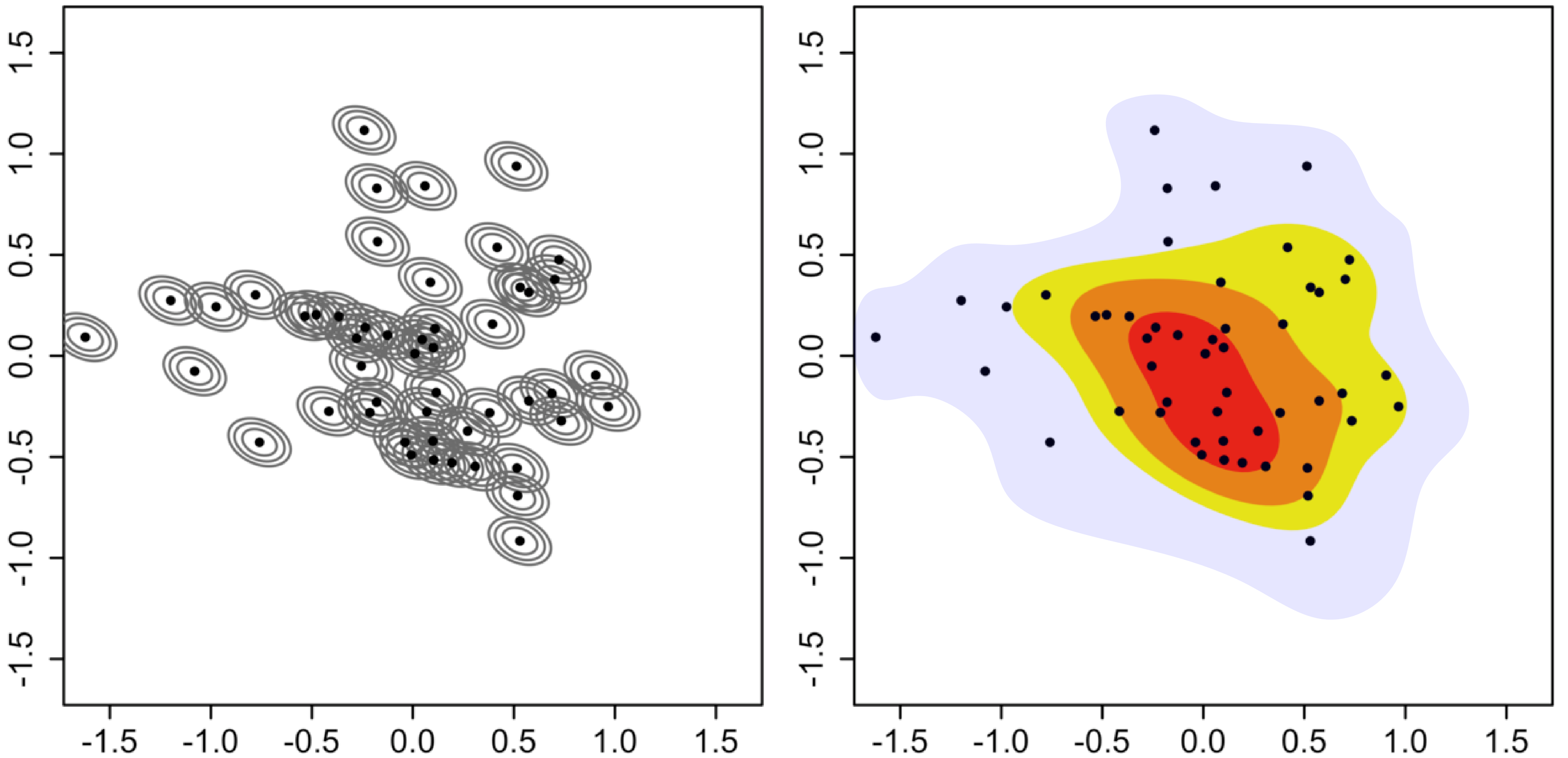
Kernel density estimators

Principle: take a mixture of copies of an 'elementary' density (kernel), anchored at each observation



Kernel density estimators

Principle: take a mixture of copies of an ‘elementary’ density (kernel), anchored at each observation



(image source: `http://www.wikiwand.com/en/Multivariate_kernel_density_estimation`)

Kernel density estimators

Input: $P = \{p_1, \cdots, p_n\} \subset \mathbb{R}^d$ (data points), $x \in \mathbb{R}^d$ (query point)

General formula: (convolution)

$$\hat{f}_{K_H}(x) := \frac{1}{n} \sum_{i=1}^{n} K_H(x - p_i), \text{ where } K_H(u) := (\det H)^{-1/2} K(H^{-1/2} u)$$

- H: inner-product (positive-definite) $d \times d$ matrix (adds scaling / anisotropy)
- $K : \mathbb{R}^d \to \mathbb{R}^+$: d-variate kernel:

$$\int_{\mathbb{R}^d} K(u)\, du = 1 \quad \text{(normalized)}$$

$$\int_{\mathbb{R}^d} u\, K(u)\, du = 0 \quad \text{(centered at origin)}$$

$$\lim_{\|u\| \to \infty} K(u) = 0 \quad \text{(vanishes at infinity)}$$

$$\int_{\mathbb{R}^d} uu^T\, K(u)\, du = c_K\, I_d \quad \text{(isotropic)}$$

Kernel density estimators

Specialization 1: take $H = \sigma^2 I_d$ (isotropic kernel)

bandwidth / window

Kernel density estimators

Specialization 1: take $H = \sigma^2 I_d$ (isotropic kernel)

bandwidth / window

Specialization 2: take $K(u) \propto k(\|u\|_2^2)$ for some $k : \mathbb{R}^+ \to \mathbb{R}^+$

(radially-symmetric kernel)

kernel profile

normalizing factor: $c_{k,d} := \left(\int_{\mathbb{R}^d} k(\|u\|_2^2)\, du \right)^{-1}$

Kernel density estimators

Specialization 1: take $H = \sigma^2 I_d$ (isotropic kernel)

bandwidth / window

Specialization 2: take $K(u) \propto k(\|u\|_2^2)$ for some $k : \mathbb{R}^+ \to \mathbb{R}^+$

(radially-symmetric kernel)

kernel profile

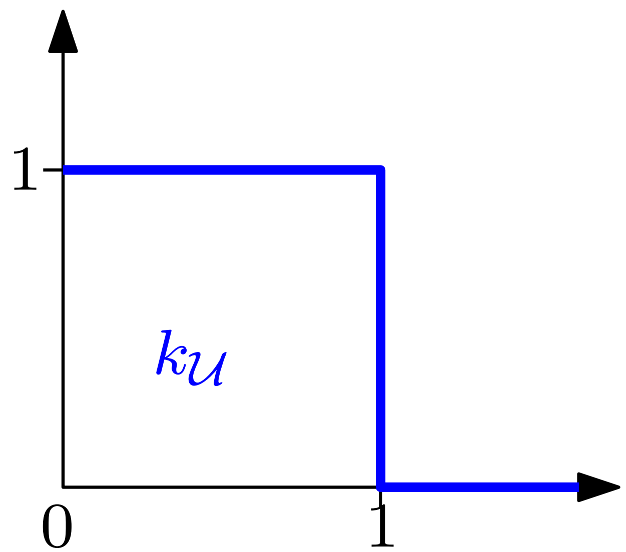
normalizing factor: $c_{k,d} := \left(\int_{\mathbb{R}^d} k(\|u\|_2^2)\, du \right)^{-1}$

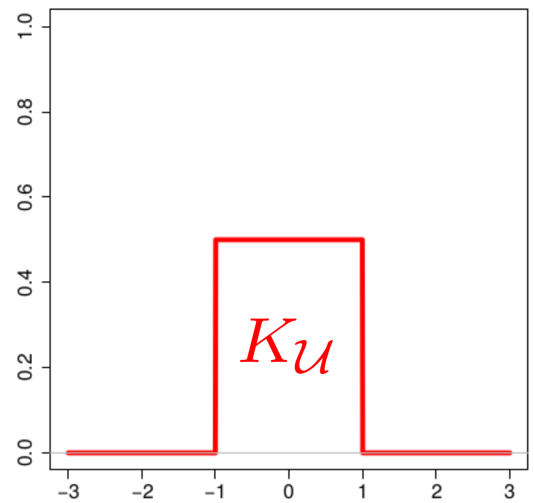
$$\rightsquigarrow \; \hat{f}_{\sigma,k}(x) := \frac{c_{k,d}}{n\,\sigma^d} \sum_{i=1}^{n} k\left(\frac{\|x - p_i\|_2^2}{\sigma^2} \right)$$

Common kernels

Flat / Uniform: $k_{\mathcal{U}}(t) := \begin{cases} 1 \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$ $\quad \rightsquigarrow c_{k,d} = 1/\text{Vol}\, B_d(0,1)$

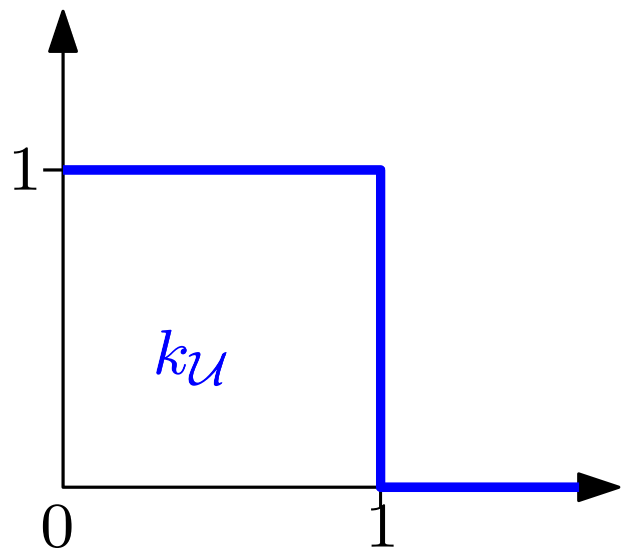
$$= \frac{\Gamma(d/2+1)}{\pi^{d/2}}$$

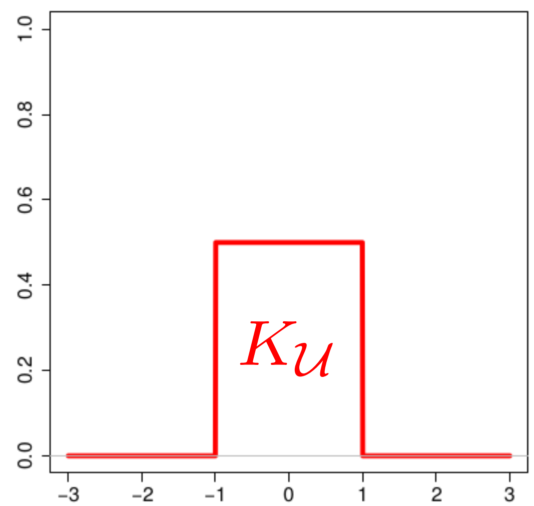




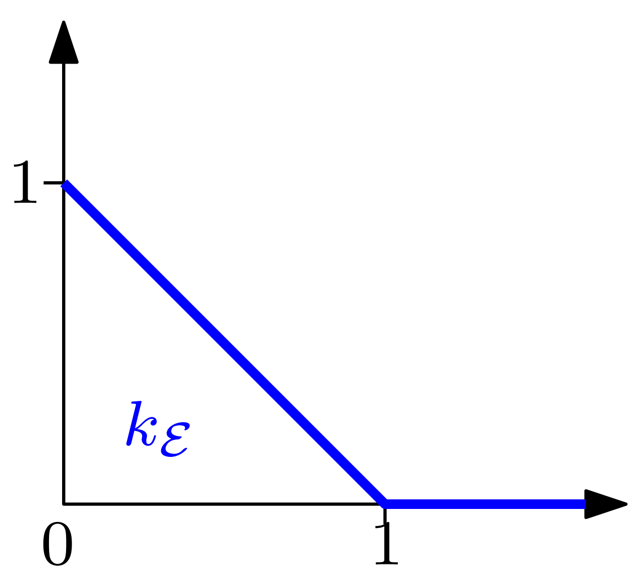
Common kernels

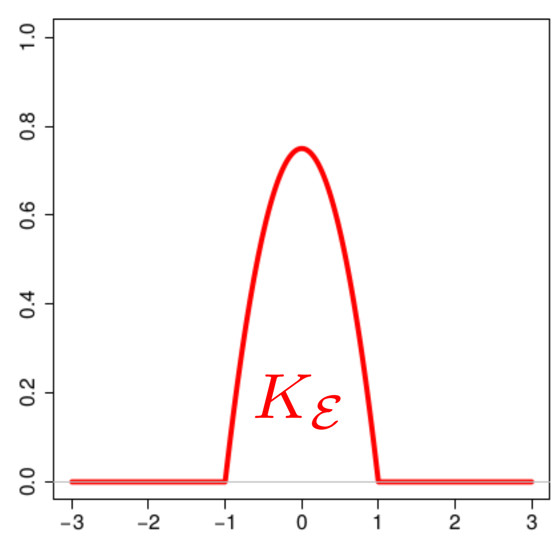
Flat / Uniform: $k_{\mathcal{U}}(t) := \begin{cases} 1 \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$ $\rightsquigarrow c_{k,d} = 1/\text{Vol}\, B_d(0,1) = \dfrac{\Gamma(d/2+1)}{\pi^{d/2}}$





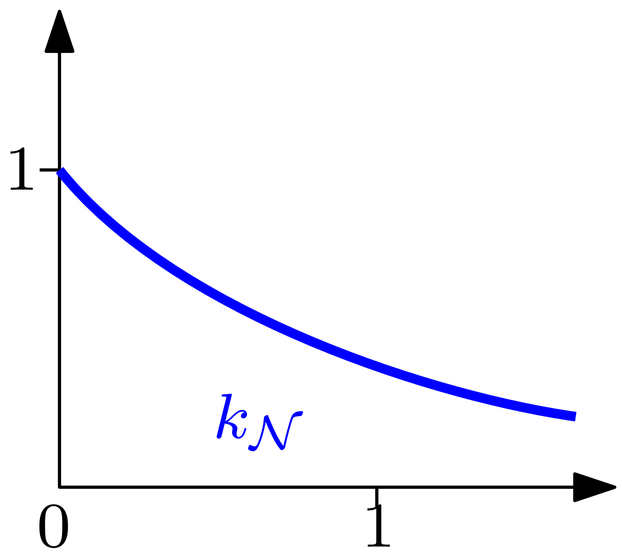
Epanechnikov: $k_{\mathcal{E}}(t) := \begin{cases} 1 - t \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$ $\rightsquigarrow c_{k,d} = \dfrac{d+2}{2\,\text{Vol}\, B_d(0,1)}$

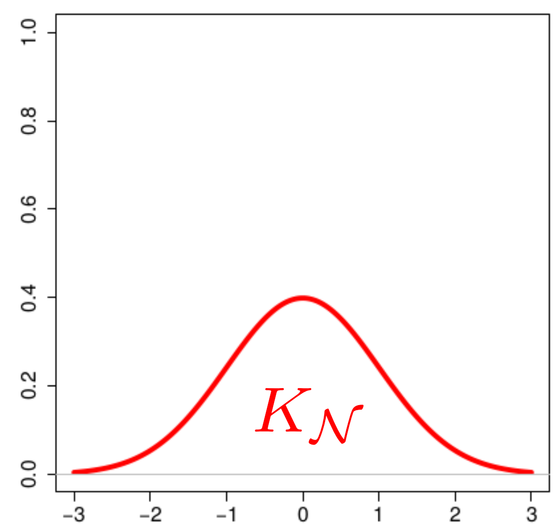




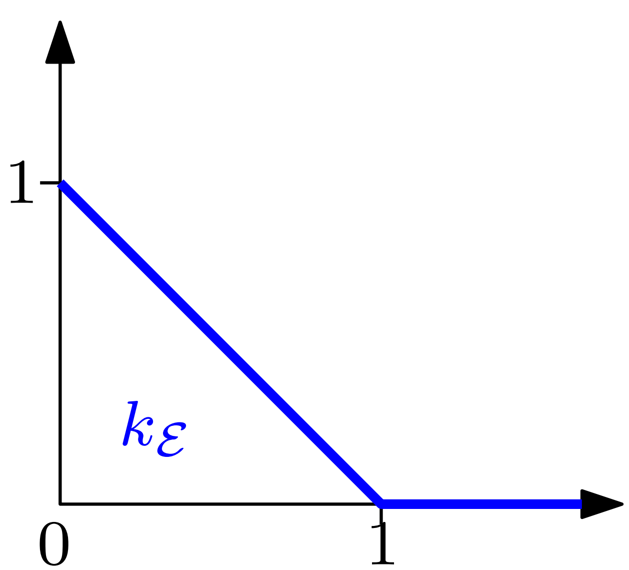
Common kernels

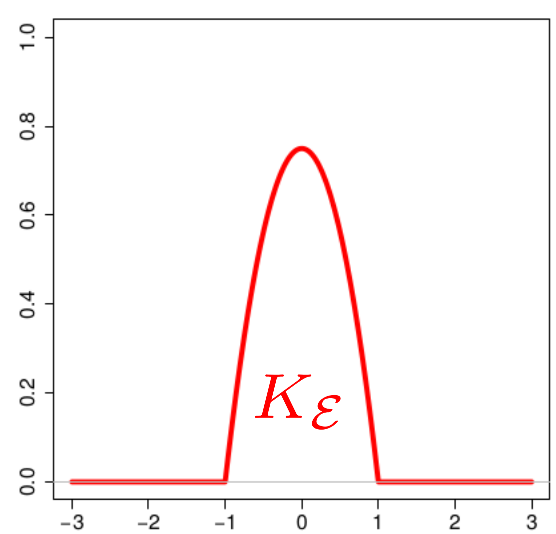
Gaussian: $k_{\mathcal{N}}(t) := \exp(-t/2)$ $\quad\rightsquigarrow c_{k,d} = (2\pi)^{-d/2}$



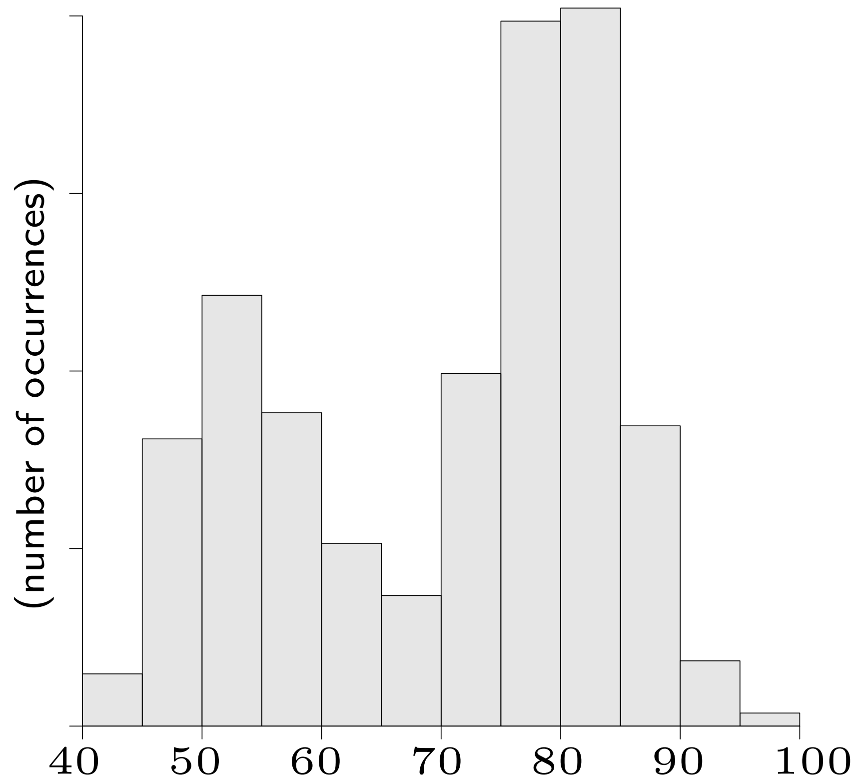


Epanechnikov: $k_{\mathcal{E}}(t) := \begin{cases} 1 - t \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$ $\quad\rightsquigarrow c_{k,d} = \dfrac{d+2}{2\,\mathrm{Vol}\,B_d(0,1)}$



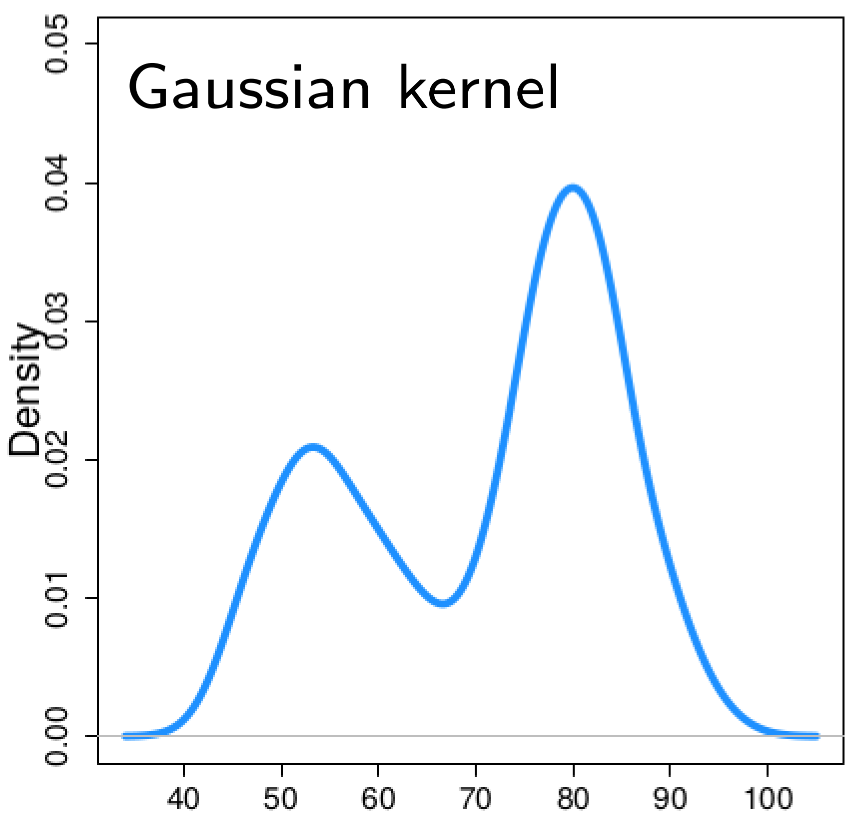


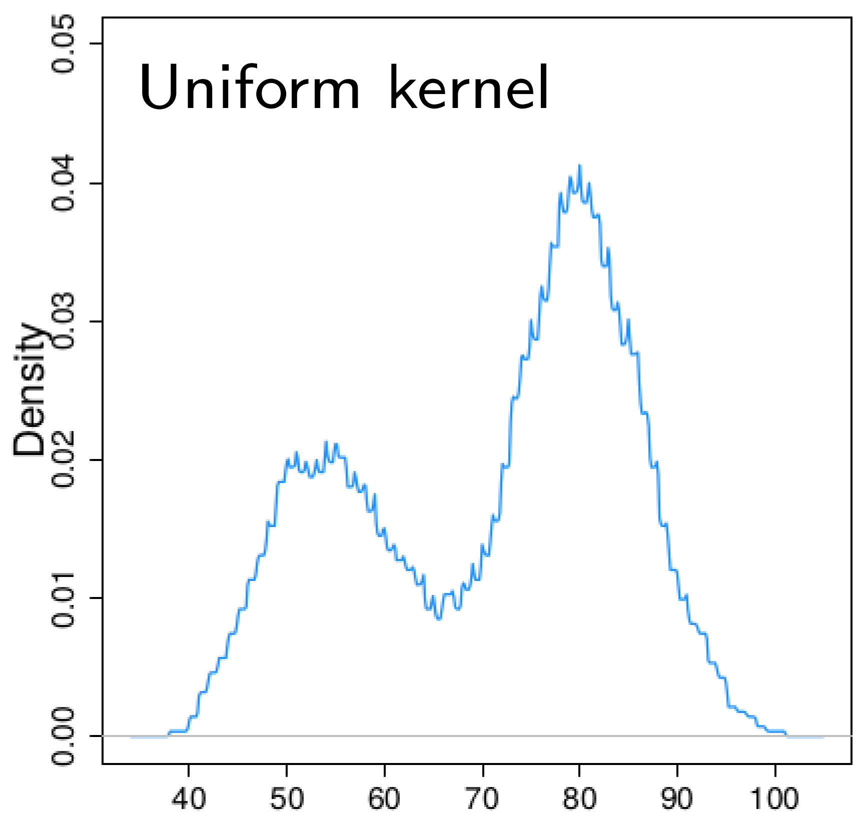
Common kernels

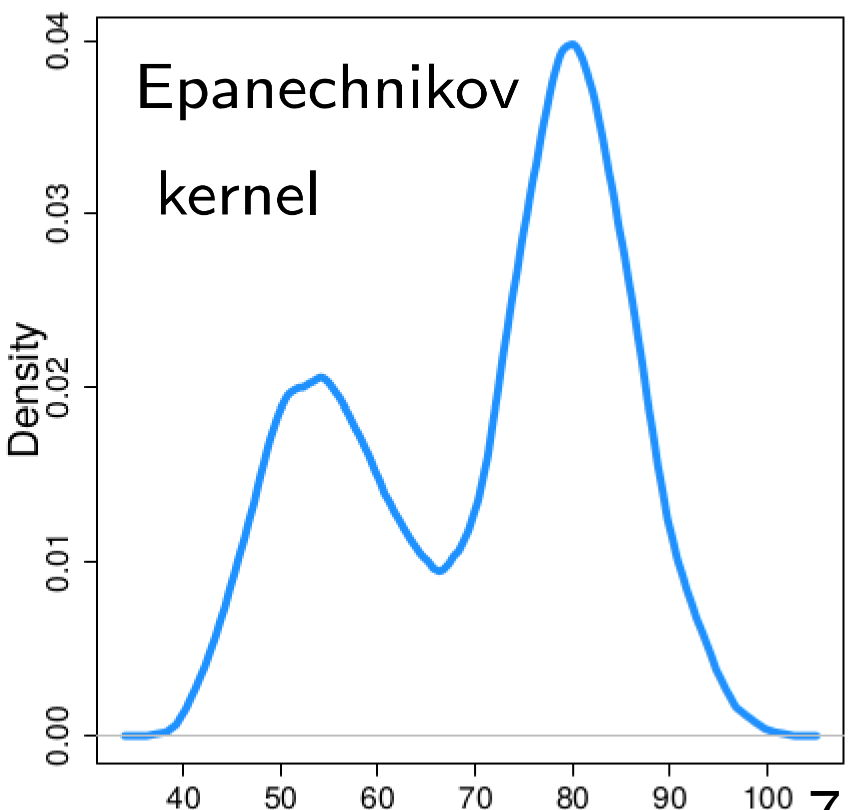


Old faithful geyser dataset (available in R):

- 1st coordinate: waiting time (sec.) between eruptions
- 2nd coordinate (unused): eruptions duration (sec.)



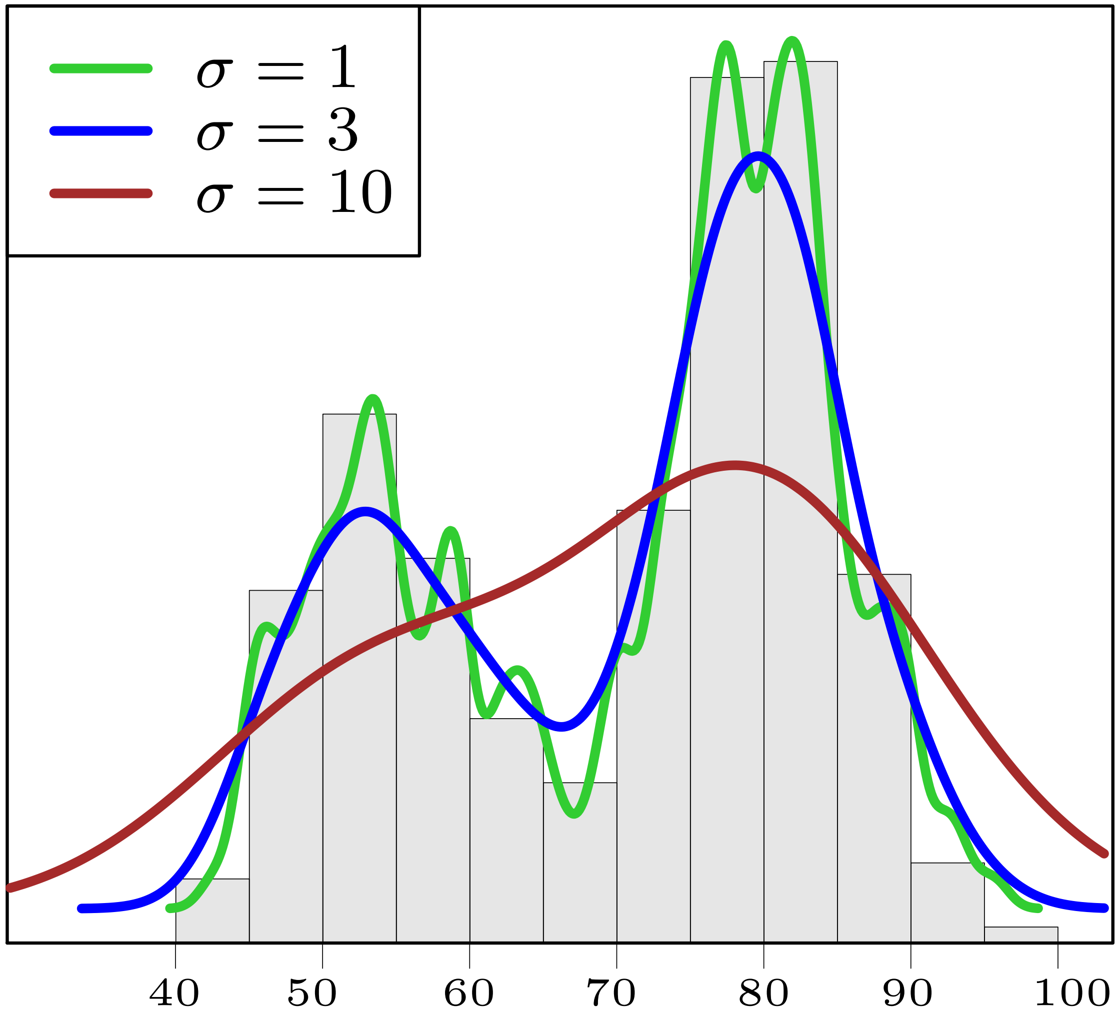




Influence of the bandwidth

- small σ (*undersmoothing*): small bias (sensitivity), large variance (instability)

- large σ (*oversmoothing*): large bias (insensitivity), small variance (stability)



Old geyser dataset

Differentiation

$$\hat{f}_{\sigma,k}(x) := \frac{c_{k,d}}{n\,\sigma^d} \sum_{i=1}^{n} k\left(\frac{\|x - p_i\|_2^2}{\sigma^2}\right)$$

$$\hat{\nabla}_f(x) := \nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \sum_{i=1}^{n} (x - p_i)\,k'\left(\frac{\|x - p_i\|_2^2}{\sigma^2}\right)$$

Differentiation

$$\hat{f}_{\sigma,k}(x) := \frac{c_{k,d}}{n\,\sigma^d} \sum_{i=1}^{n} k\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)$$

$$\hat{\nabla}_f(x) := \nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \sum_{i=1}^{n} (x-p_i)\,k'\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)$$

Letting $g := -k'$ (assumed to be ≥ 0):

$$\nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \left(\sum_{i=1}^{n} g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)\right) \left(\frac{\sum_{i=1}^{n} p_i\, g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)}{\sum_{i=1}^{n} g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)} - x\right)$$

Differentiation

$$\hat{f}_{\sigma,k}(x) := \frac{c_{k,d}}{n\,\sigma^d} \sum_{i=1}^{n} k\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)$$

$$\hat{\nabla}_f(x) := \nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \sum_{i=1}^{n} (x-p_i)\,k'\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)$$

Letting $g := -k'$ (assumed to be ≥ 0):

$$\nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \underbrace{\left(\sum_{i=1}^{n} g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)\right)}_{\text{(un-normalized) kernel density estimator with profile } g} \underbrace{\left(\underbrace{\frac{\sum_{i=1}^{n} p_i\, g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)}{\sum_{i=1}^{n} g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)}}_{\text{barycenter w.r.t. } g} - x\right)}_{\textbf{mean-shift } m_{\sigma,g}(x)}$$

(un-normalized) kernel density estimator with profile g

barycenter w.r.t. g

mean-shift $m_{\sigma,g}(x)$

Differentiation

$$\hat{f}_{\sigma,k}(x) := \frac{c_{k,d}}{n\,\sigma^d} \sum_{i=1}^{n} k\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)$$

$$\hat{\nabla}_f(x) := \nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \sum_{i=1}^{n} (x-p_i)\,k'\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)$$

Letting $g := -k'$ (assumed to be ≥ 0):

$$\nabla_{\hat{f}_{\sigma,k}}(x) = \frac{2\,c_{k,d}}{n\,\sigma^{d+2}} \underbrace{\left(\sum_{i=1}^{n} g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)\right)} \underbrace{\left(\underbrace{\frac{\sum_{i=1}^{n} p_i\, g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)}{\sum_{i=1}^{n} g\left(\frac{\|x-p_i\|_2^2}{\sigma^2}\right)}} - x\right)}$$

(un-normalized) kernel density estimator with profile g

barycenter w.r.t. g

mean-shift $m_{\sigma,g}(x)$

$\Rightarrow$ gradient of density is collinear with mean-shift and oriented in the same direction

Mean-Shift

hill-climbing

Input: $P = \{p_1, \cdots, p_n\} \subset \mathbb{R}^d$ (data points), $x \in \mathbb{R}^d$ (query point to be labeled)

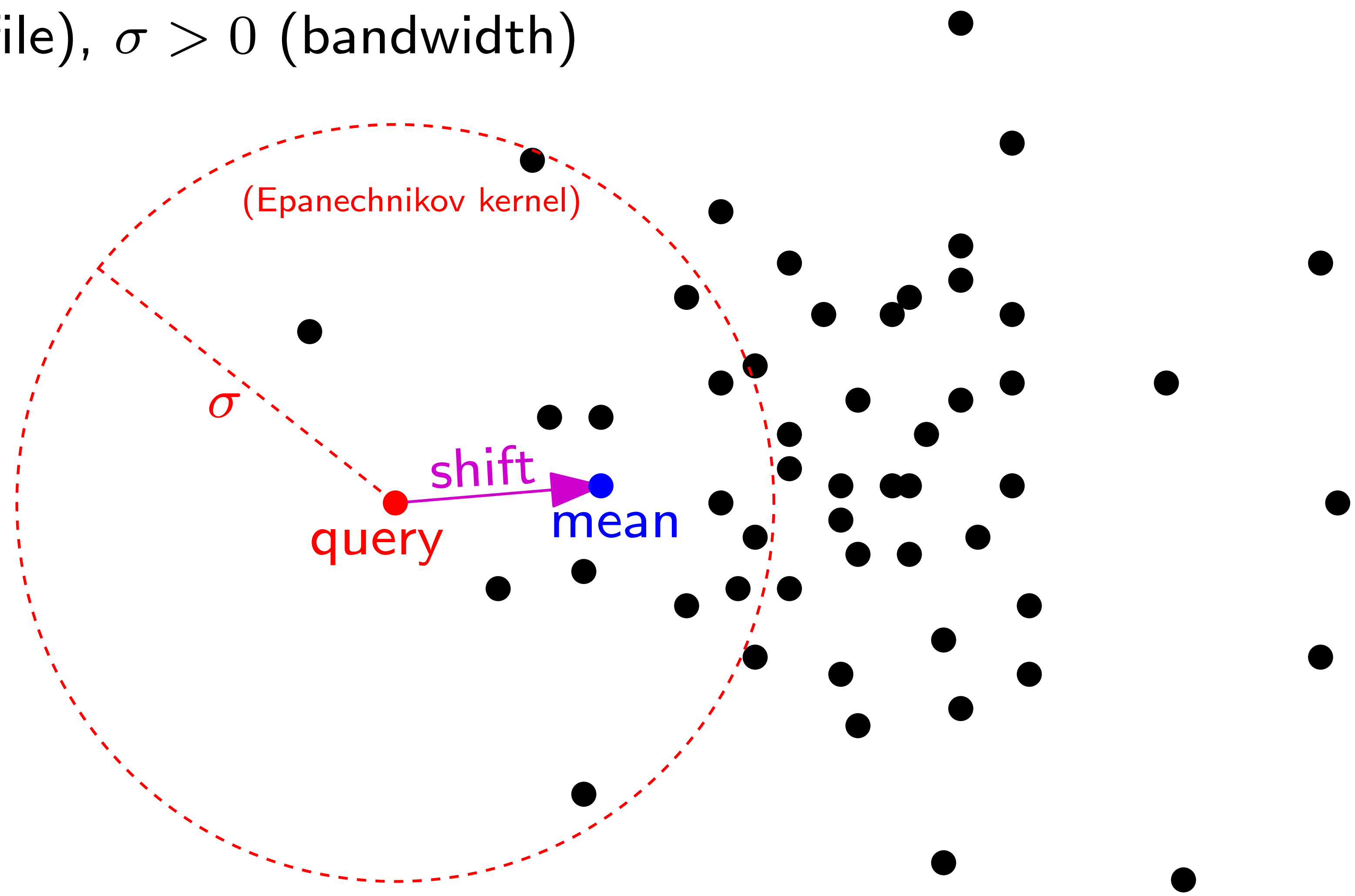
Parameters: $k\colon \mathbb{R}^+ \to \mathbb{R}^+$ (profile), $\sigma > 0$ (bandwidth)

$x_0 := x$

Repeat:

$x_{j+1} := x_j + m_{\sigma,g}(x_j)$

until convergence



Output: the label associated with the convergence point

Mean-Shift

- Apply Mean-Shift hill-climbing to each input point $p_i \in P$
- Epanechnikov kernel $\Rightarrow$ convergence in finite time

  $\rightarrow$ may converge outside the set of critical points of the estimator

  $\rightarrow$ use variant to guarantee cvgence to maximum [Huang et al. 2017]

Mean-Shift

- Apply Mean-Shift hill-climbing to each input point $p_i \in P$
- Epanechnikov kernel $\Rightarrow$ convergence in finite time

  $\rightarrow$ may converge outside the set of critical points of the estimator

  $\rightarrow$ use variant to guarantee cvgence to maximum [Huang et al. 2017]
- Gaussian kernel $\Rightarrow$ convergence at the limit (infinite time)

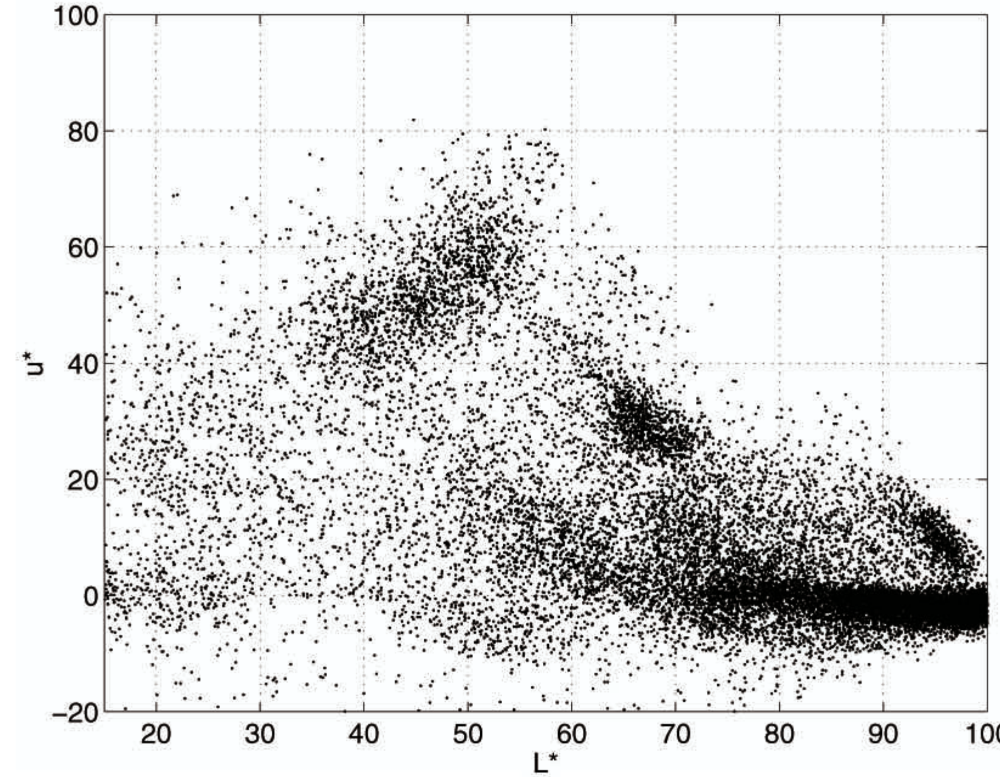
  $\rightarrow$ stopping criterion (convergence radius)

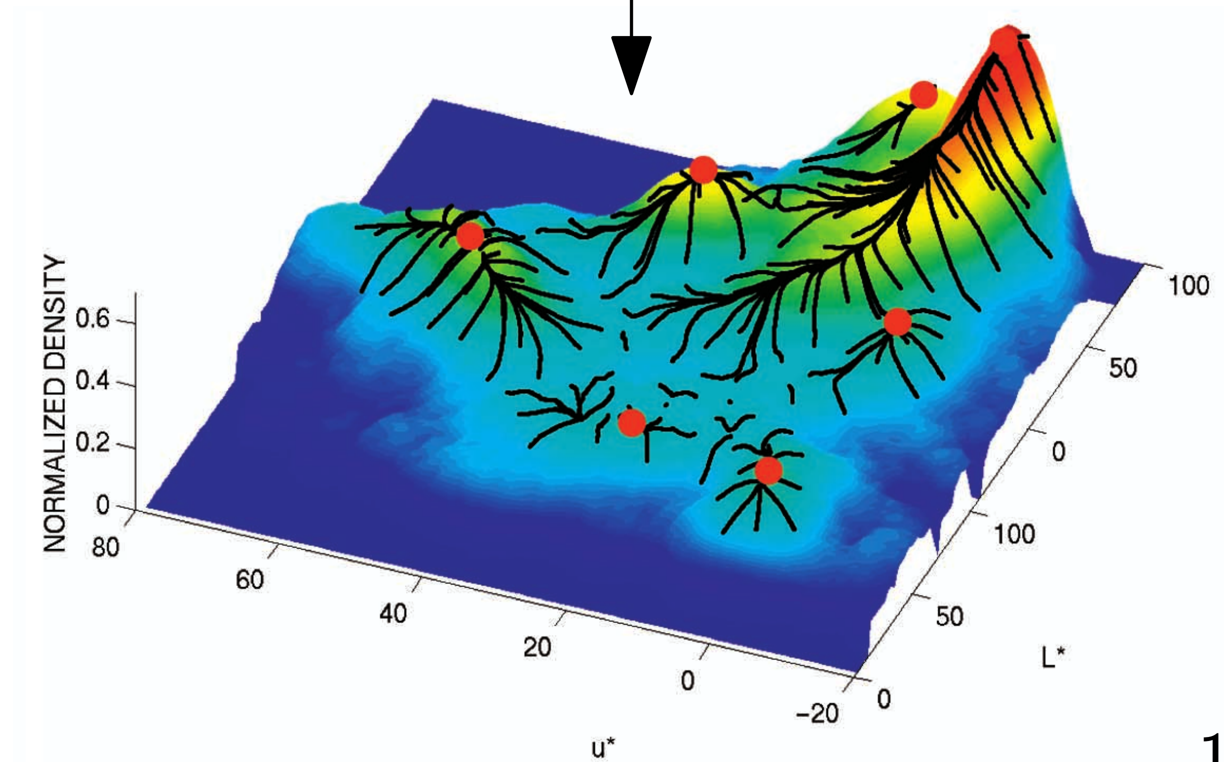
  $\rightarrow$ identification of modes (mode radius)

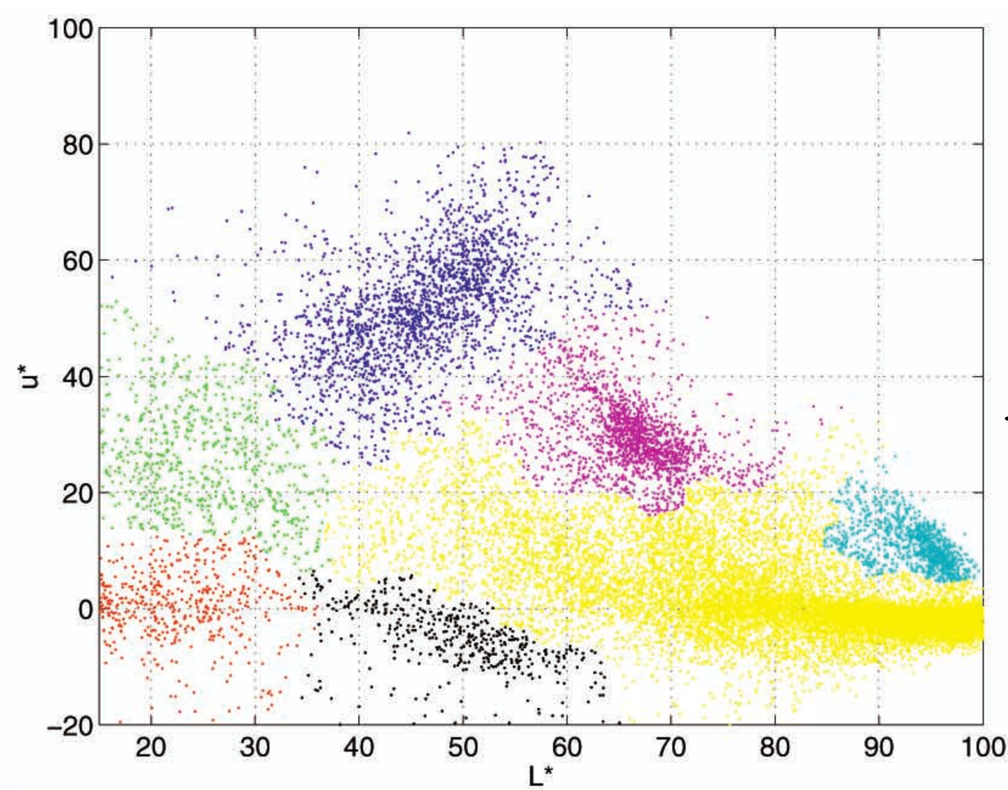
  $\rightarrow$ speed-up: hill-climbing gathers neighboring points (gathering radius)

$\rightsquigarrow$ heuristic: make these radii proportional to the estimator's bandwidth σ

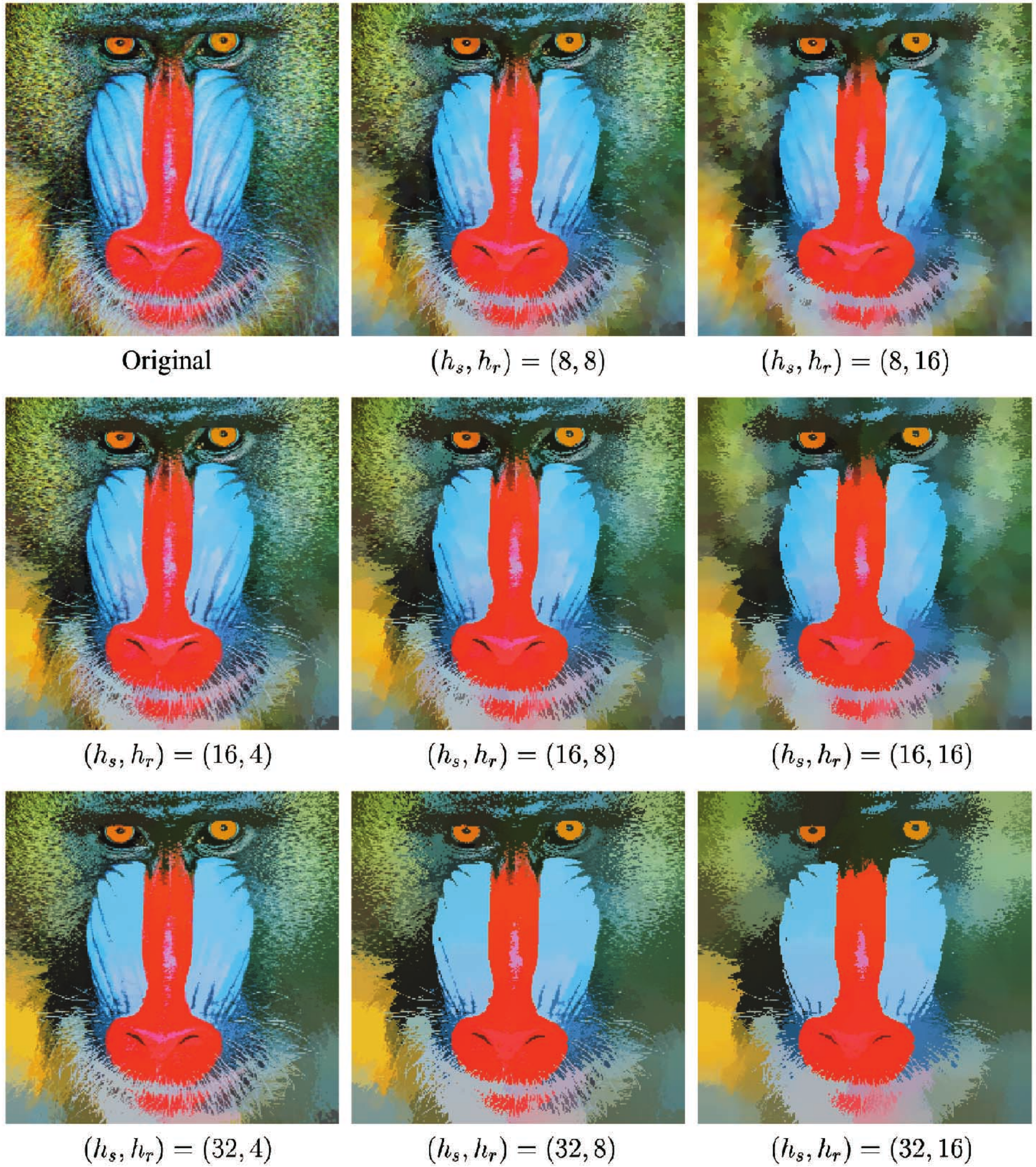
Examples [Comaniciu, Meer 2002]





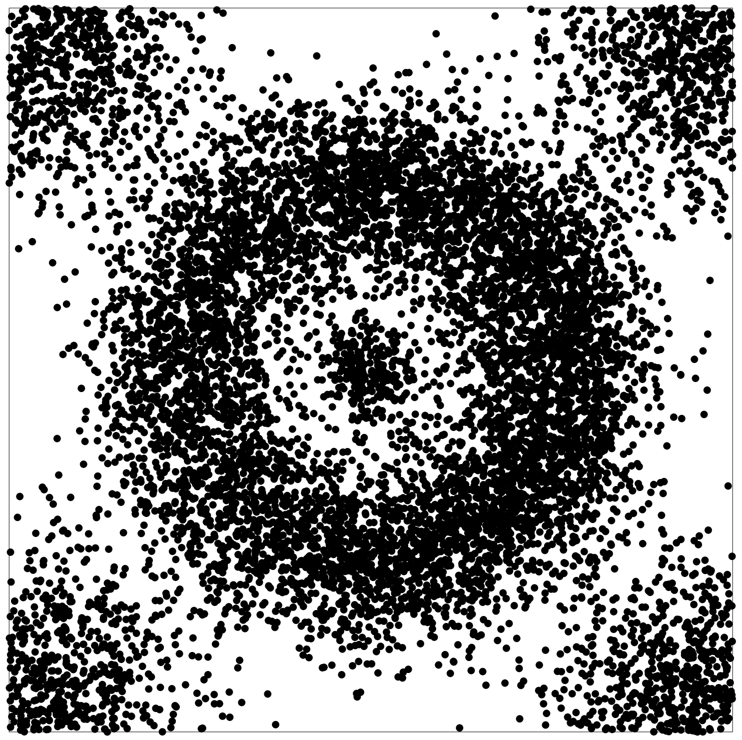


Examples [Comaniciu, Meer 2002]

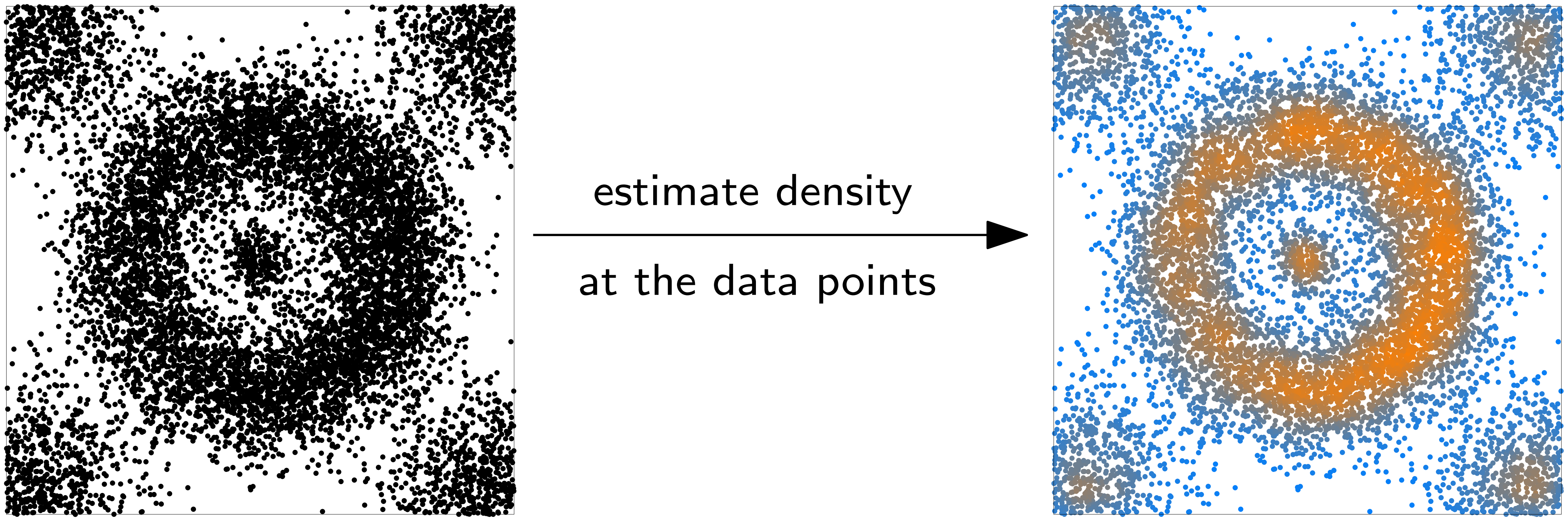


2. ToMATo

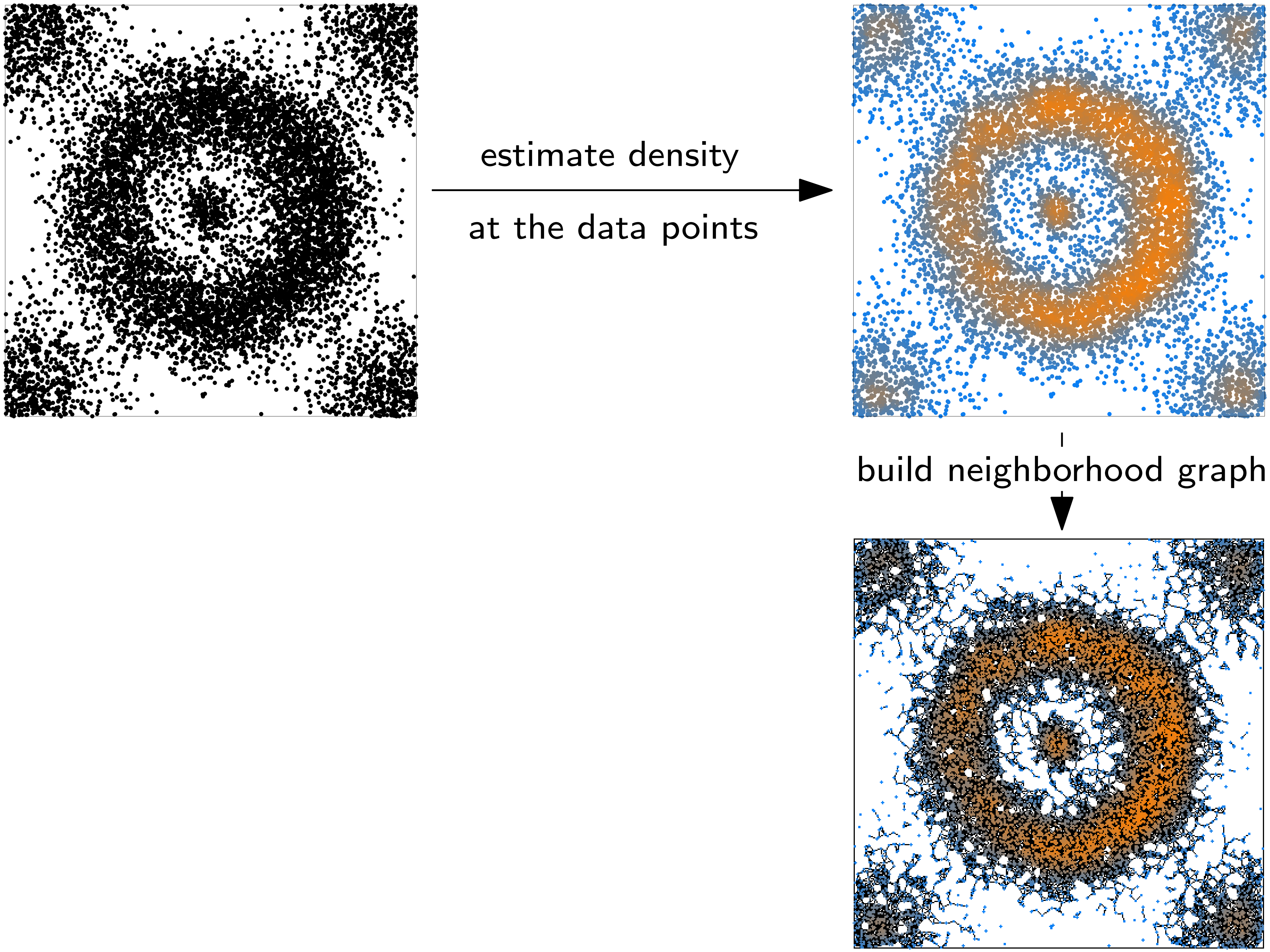
[Koontz, Narendra, Fukunaga'76] in a Nutshell



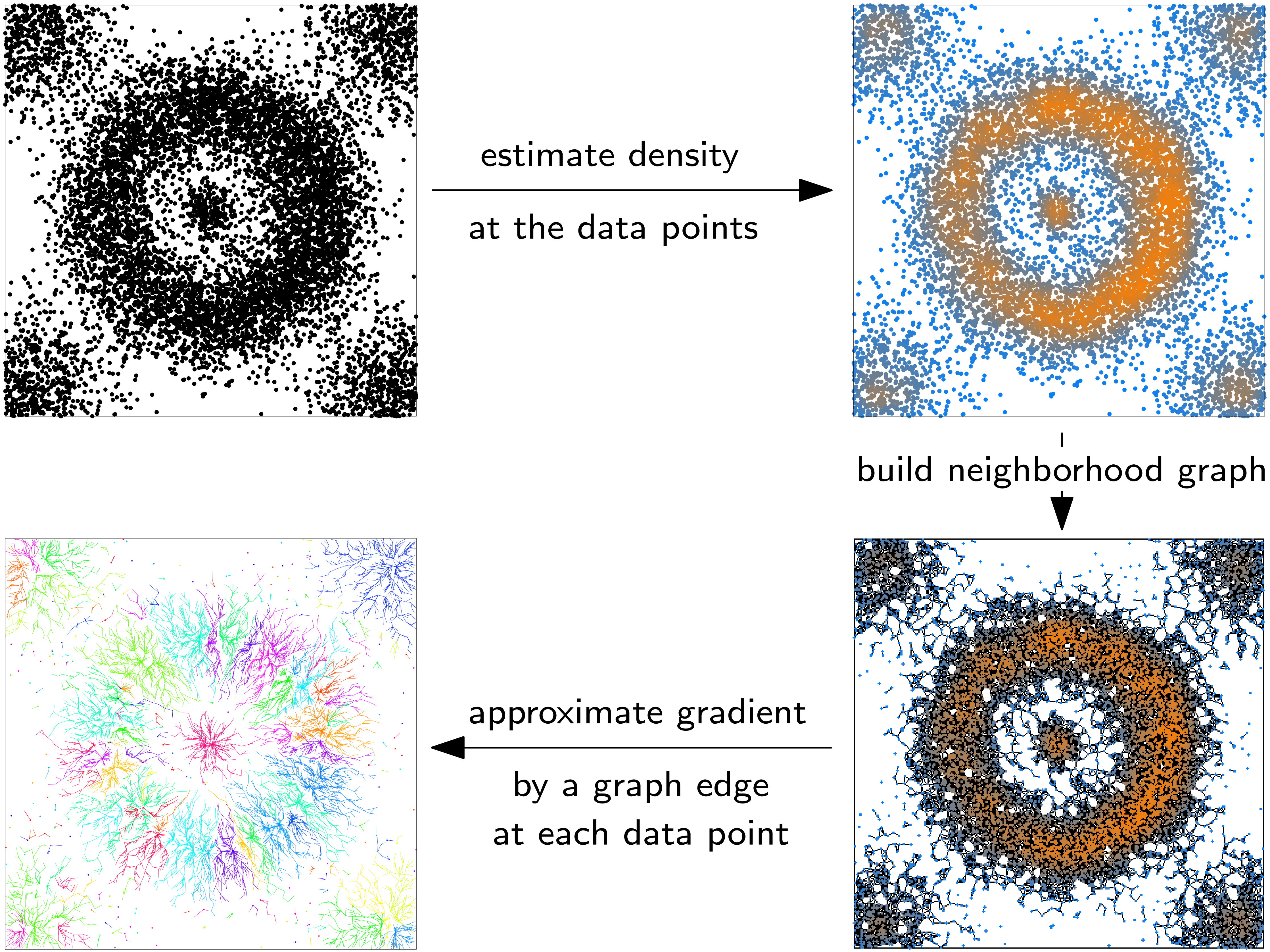
[Koontz, Narendra, Fukunaga'76] in a Nutshell



[Koontz, Narendra, Fukunaga'76] in a Nutshell



[Koontz, Narendra, Fukunaga'76] in a Nutshell



Pseudo-code:

Input: neighborhood graph G with n vertices, n-dimensional vector $\hat{f}$ (density estimator)

Sort the vertex indices $\{1, 2, \cdots, n\}$ so that $\hat{f}(1) \geq \hat{f}(2) \geq \cdots \geq \hat{f}(n)$;
Initialize a union-find data structure (disjoint-set forest) $\mathcal{U}$ and two vectors g, r of size n;

for $i = 1$ to n **do**
  Let $\mathcal{N}$ be the set of neighbors of i in G that have indices lower than i;
  if $\mathcal{N} = \emptyset$ *// vertex i is a peak of $\hat{f}$ within G*
    Create a new entry e in $\mathcal{U}$ and attach vertex i to it;
    $r(e) \leftarrow i$ *// $r(e)$ stores the root vertex associated with the entry e*
  else *// vertex i is not a peak of $\hat{f}$ within G*
    $g(i) \leftarrow \mathrm{argmax}_{j \in \mathcal{N}} \hat{f}(j)$ *// $g(i)$ stores the approximate gradient at vertex i*
    $e_i \leftarrow \mathcal{U}.\mathtt{find}(g(i))$;
    Attach vertex i to the entry e_i;

graph-based hill-climbing (1976)

Output: the collection of entries e in $\mathcal{U}$

Enter Topological Persistence...

Topological Persistence (in a nutshell)

X topological space

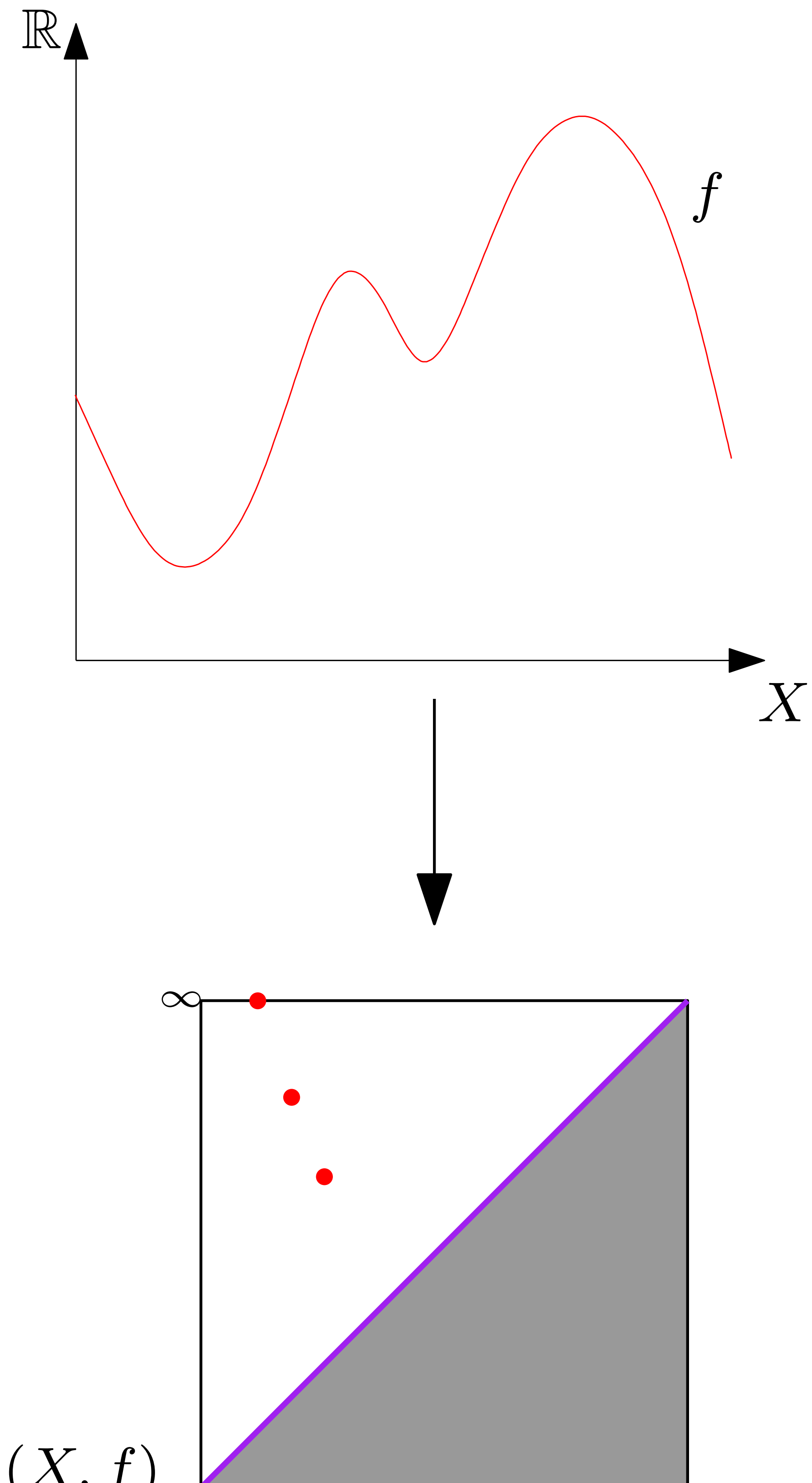
$f : X \to \mathbb{R}$

persistence

$\mathrm{Dg}\, f$

signature: *persistence diagram*

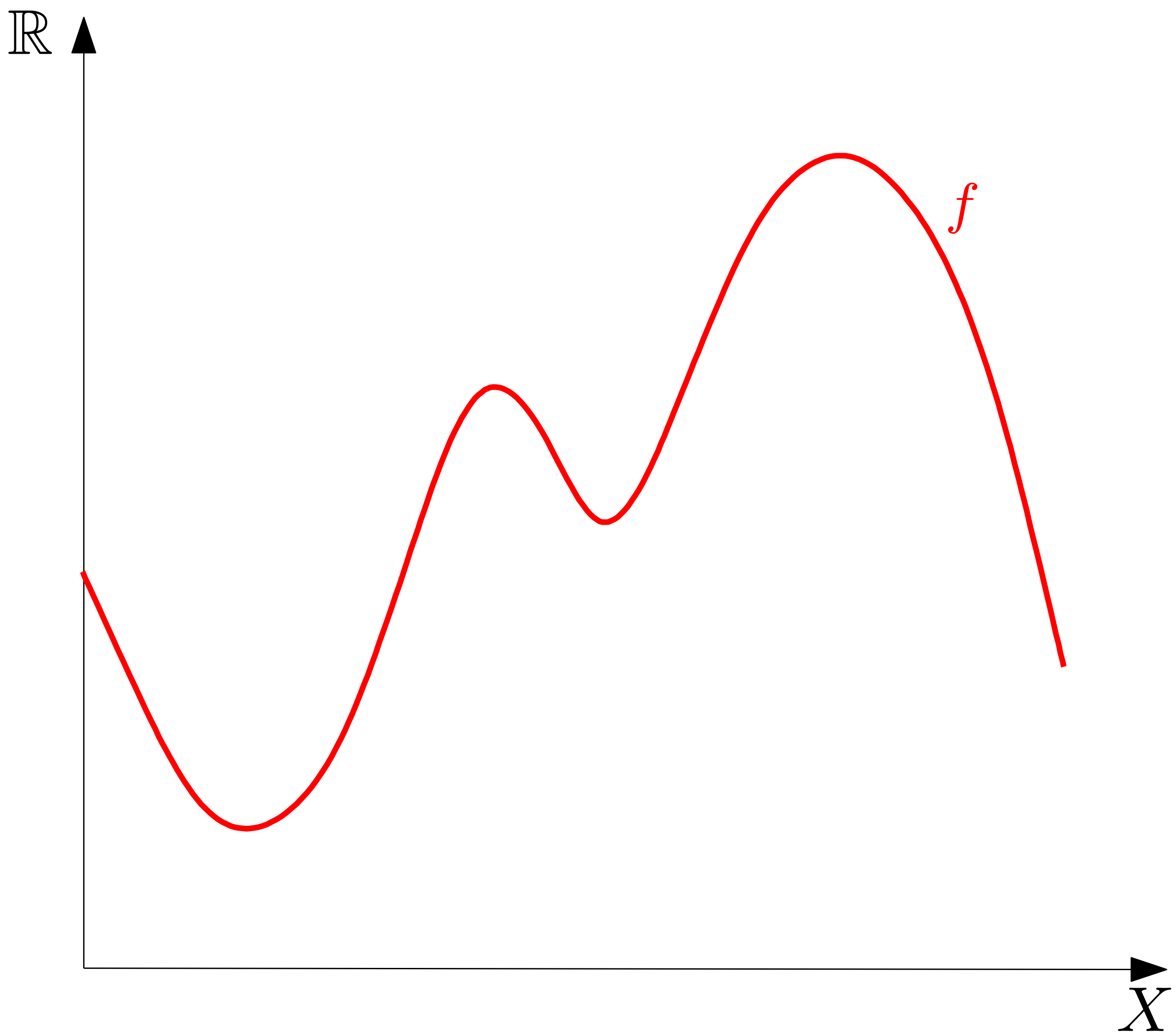
encodes the topological structure of the pair (X, f)



Topological Persistence (in a nutshell)

Inside the black box:

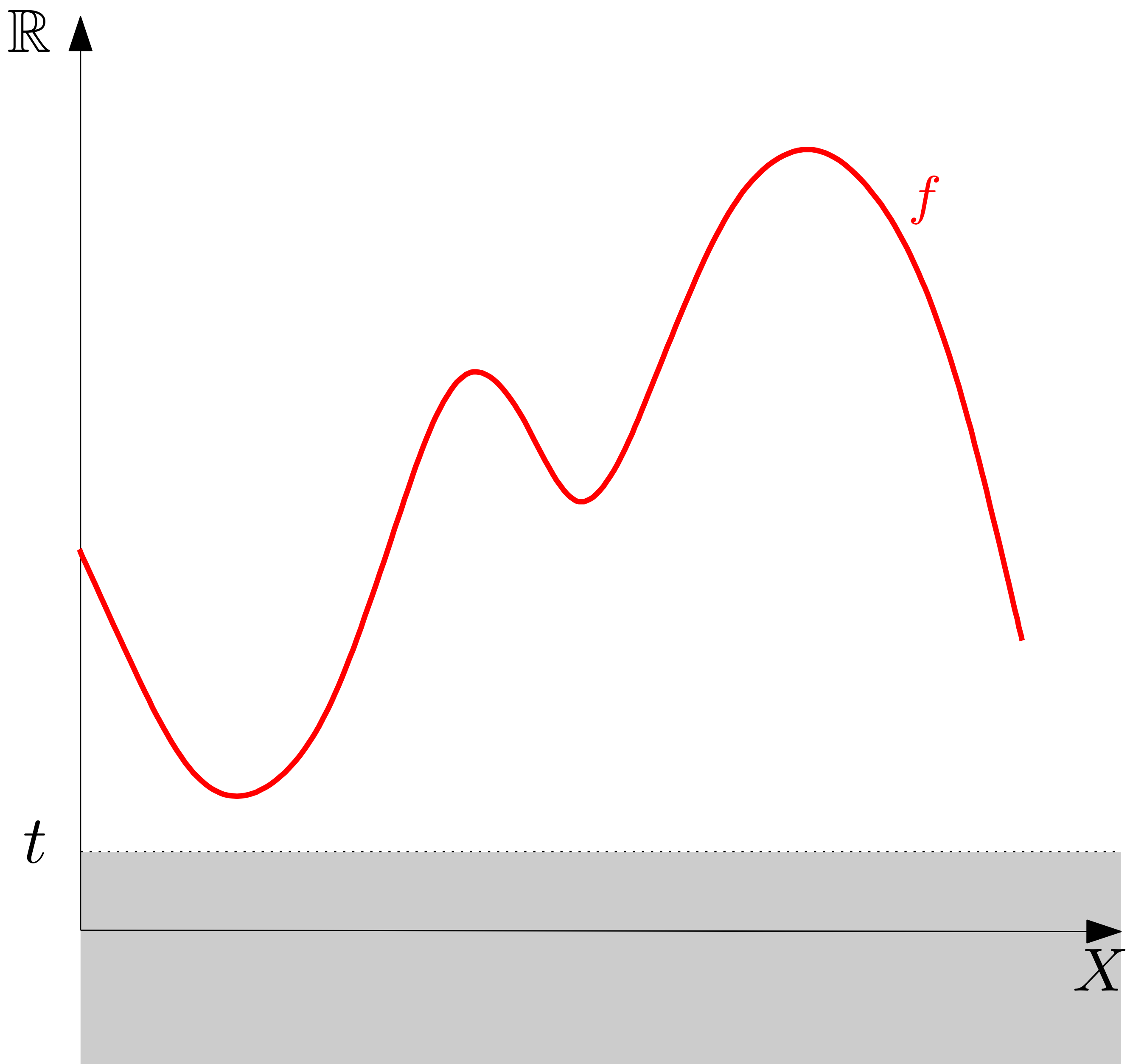
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

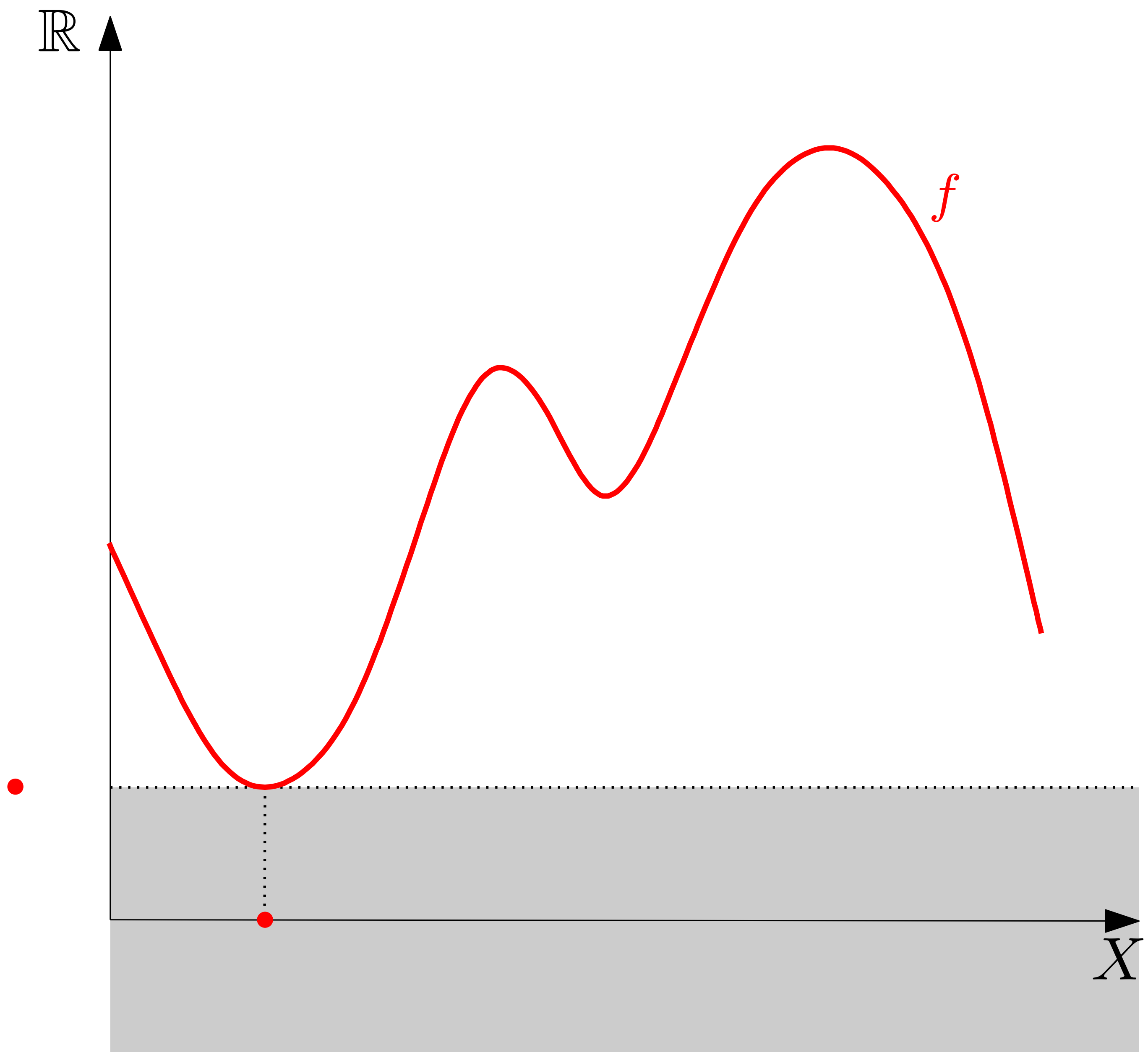
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

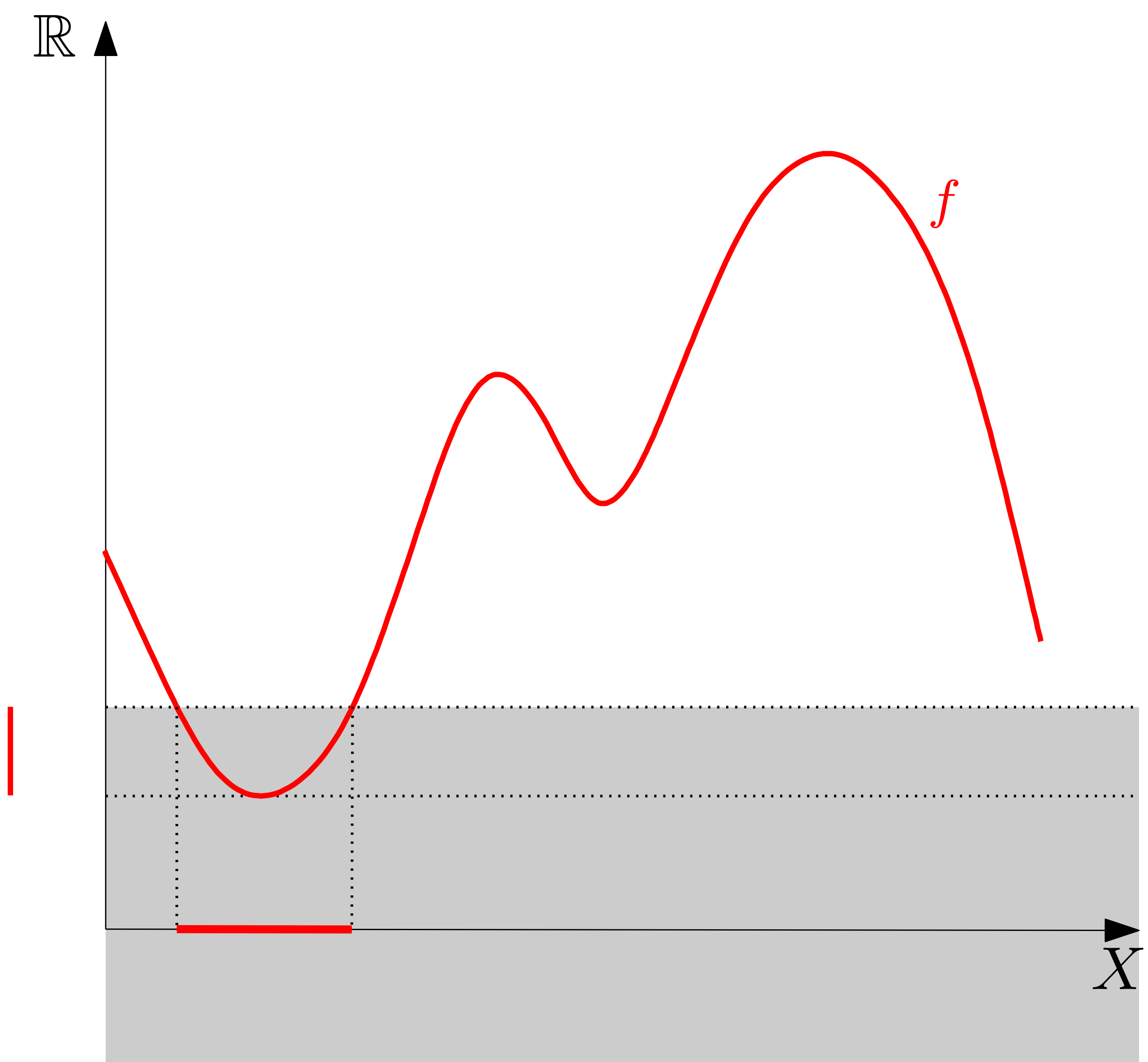
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

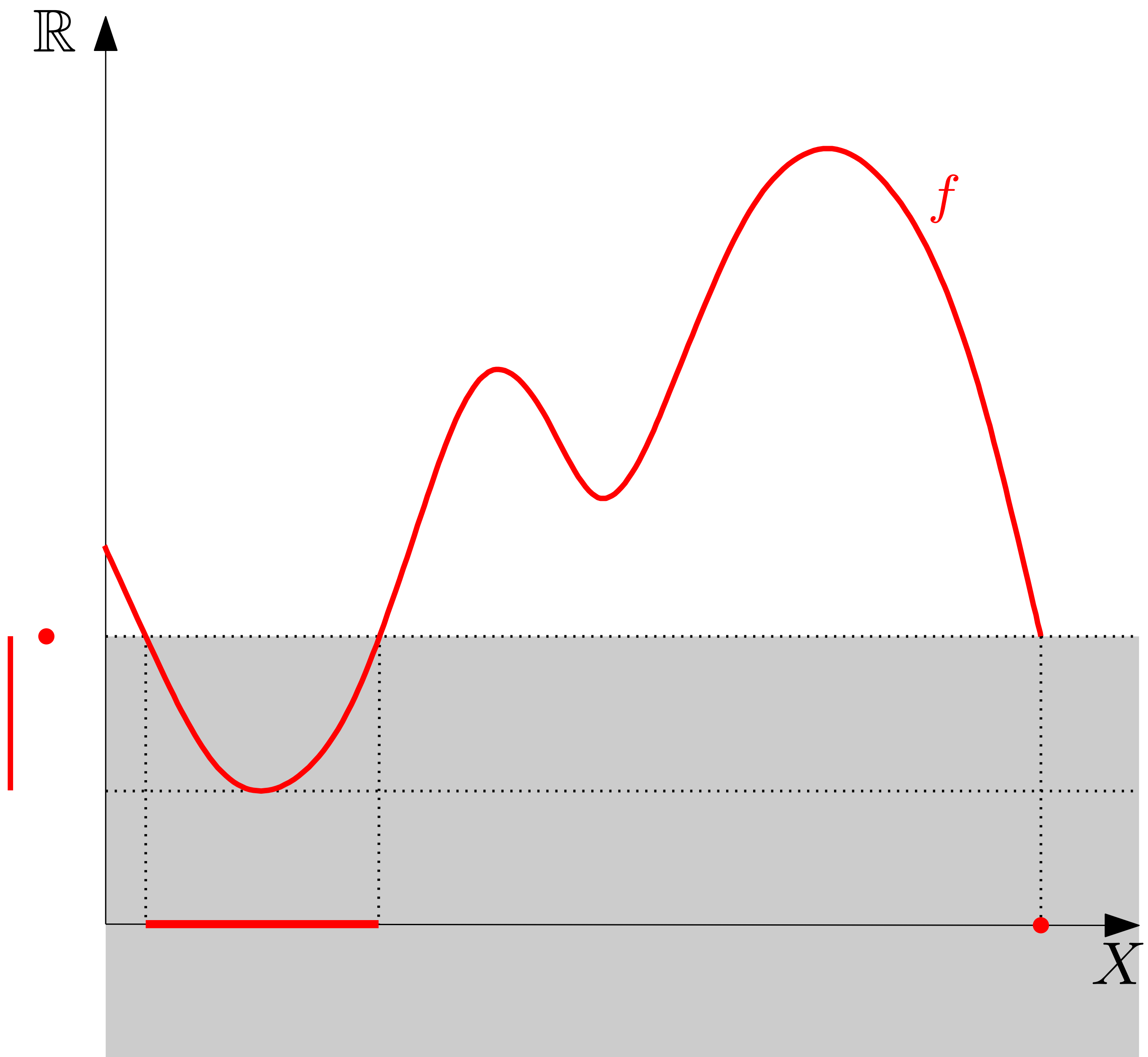
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

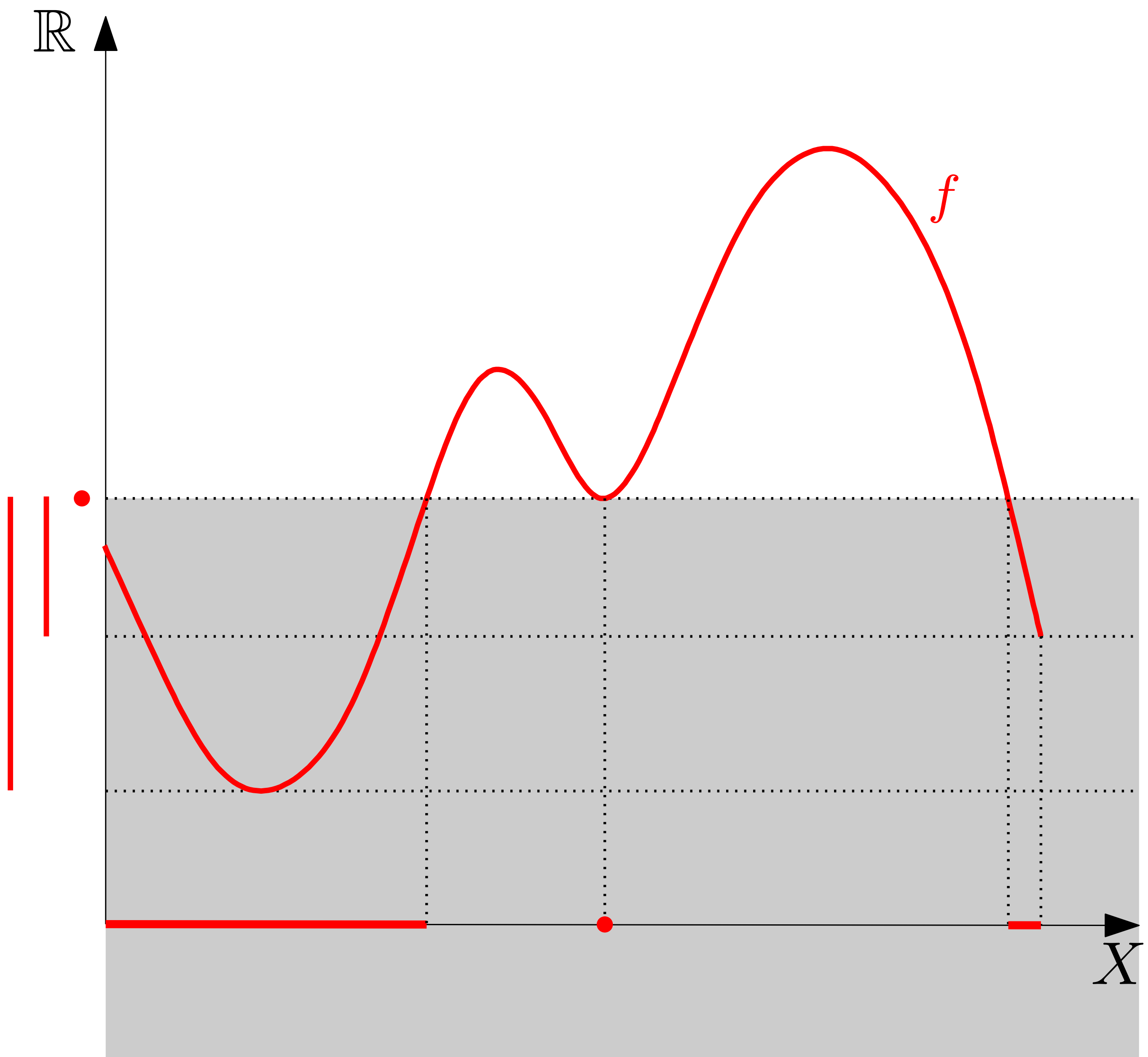
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

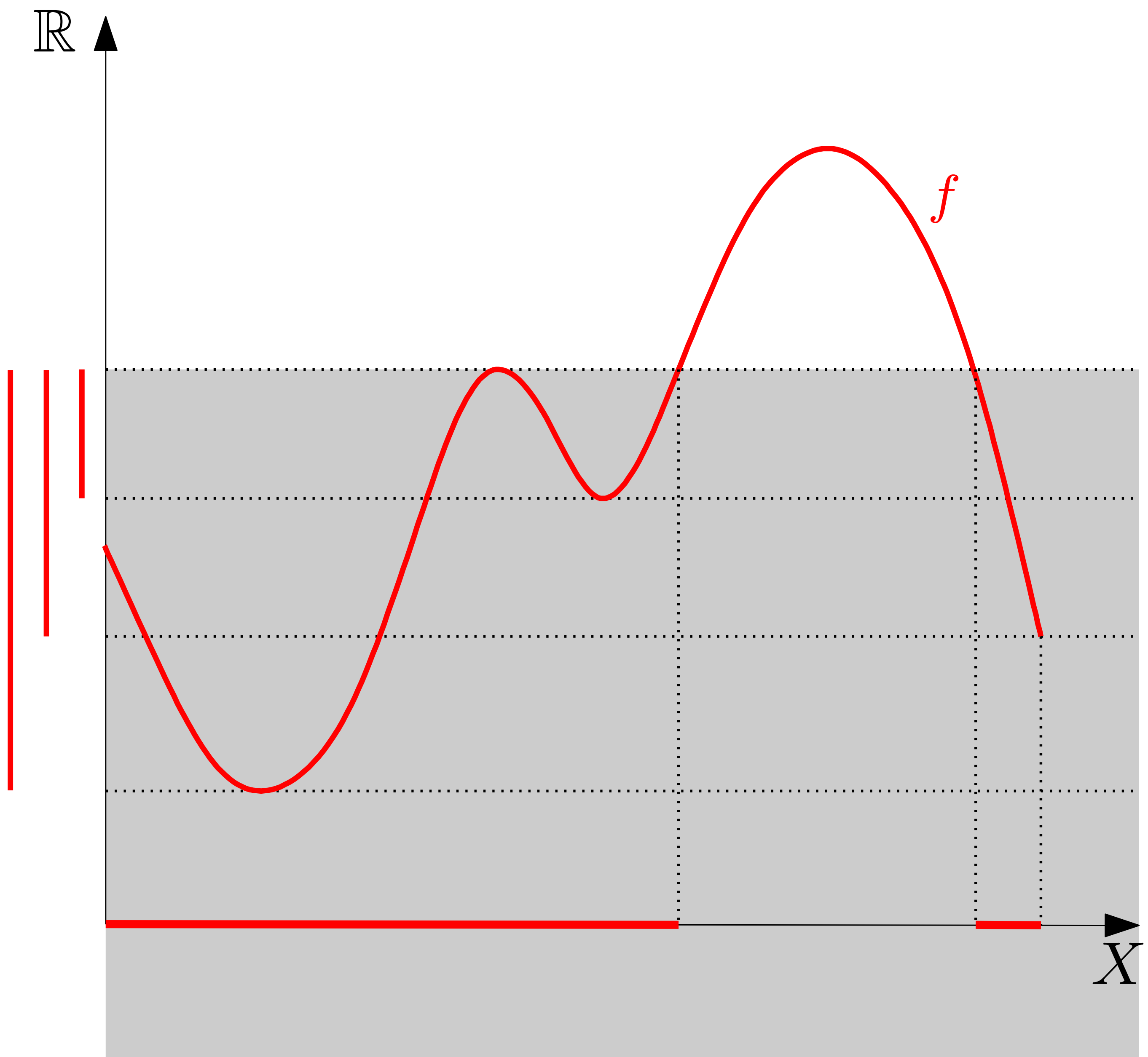
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

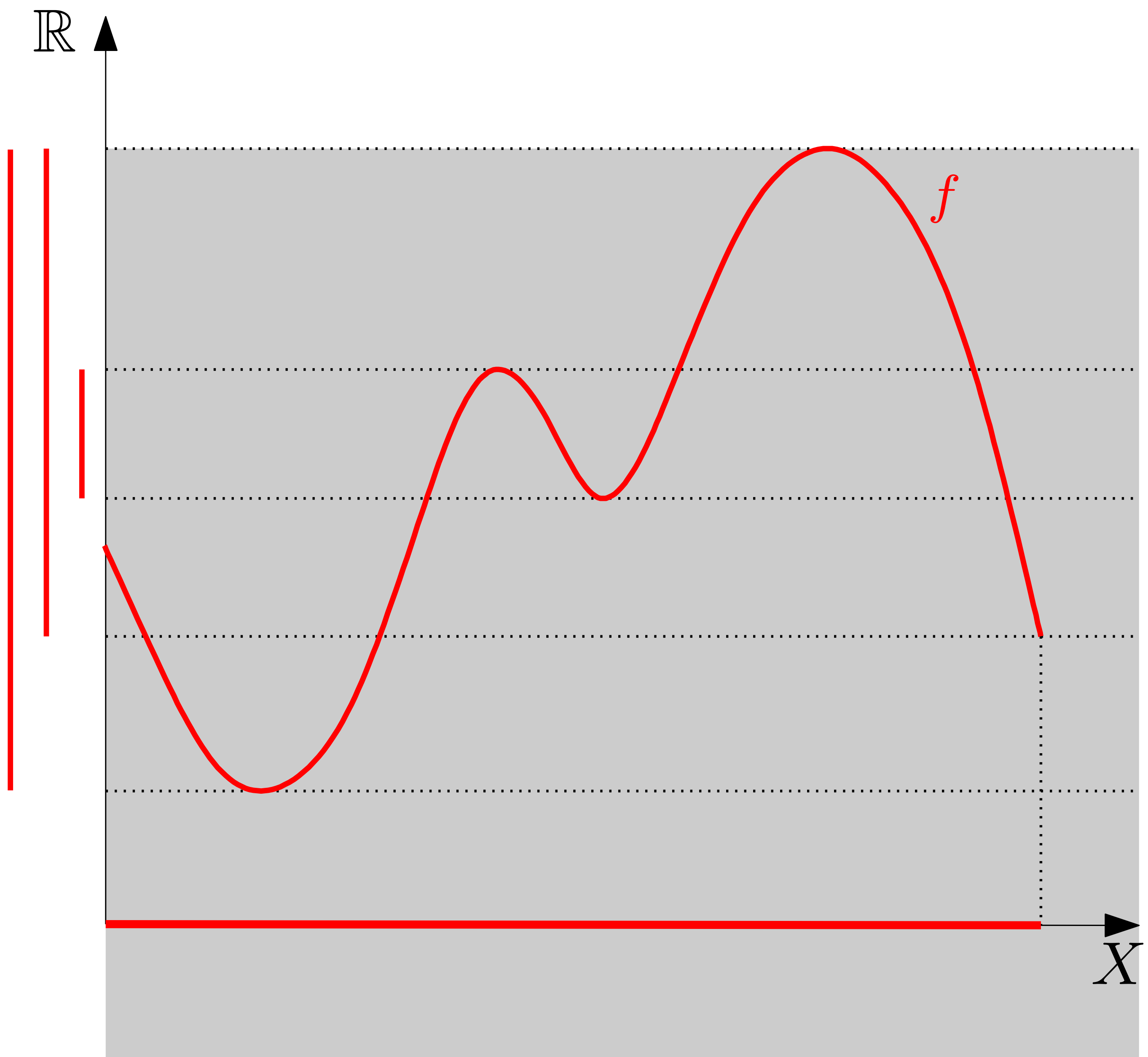
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

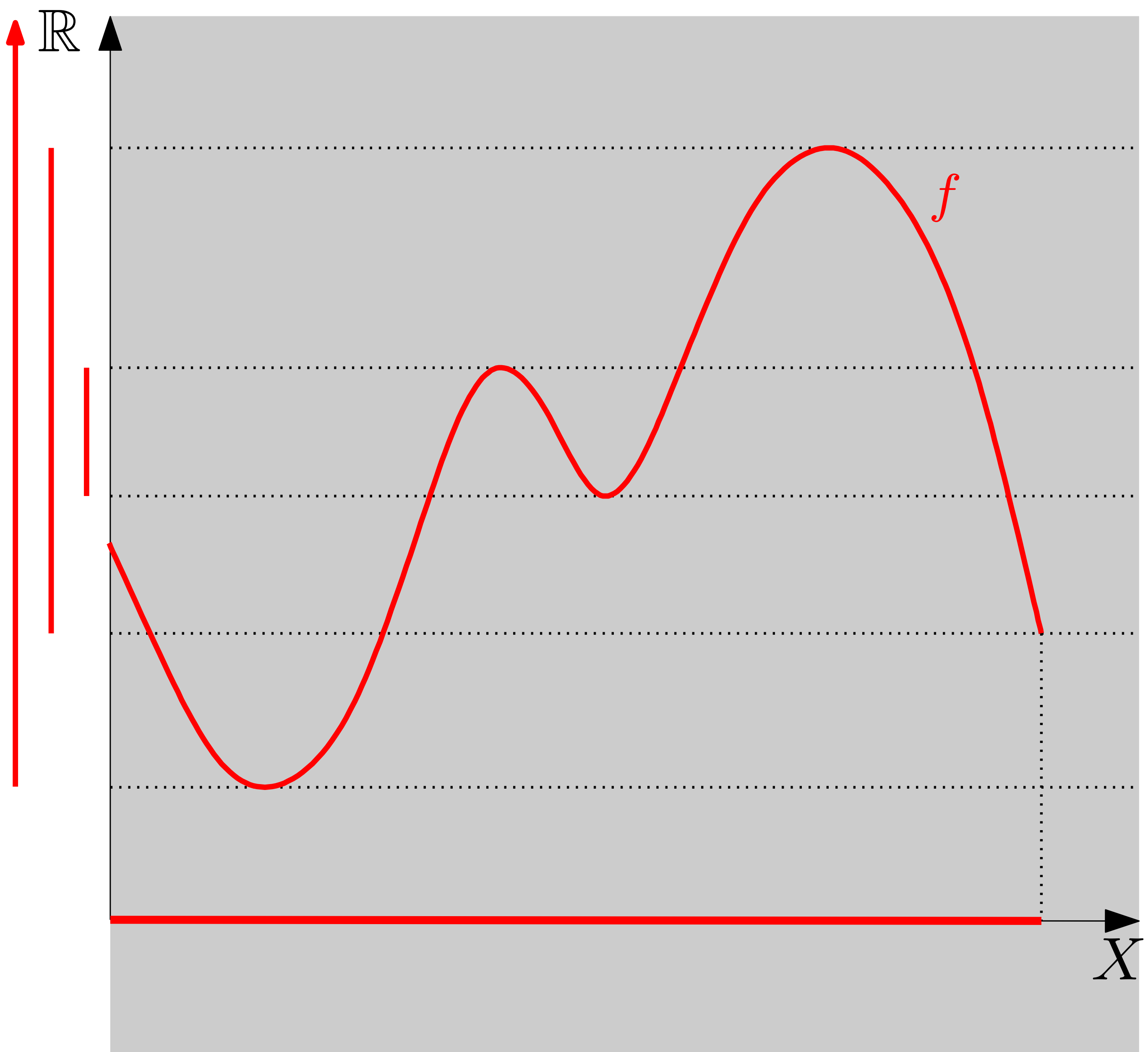
- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family



Topological Persistence (in a nutshell)

Inside the black box:

- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family
- Finite set of intervals (barcode) encodes births/deaths of topological features

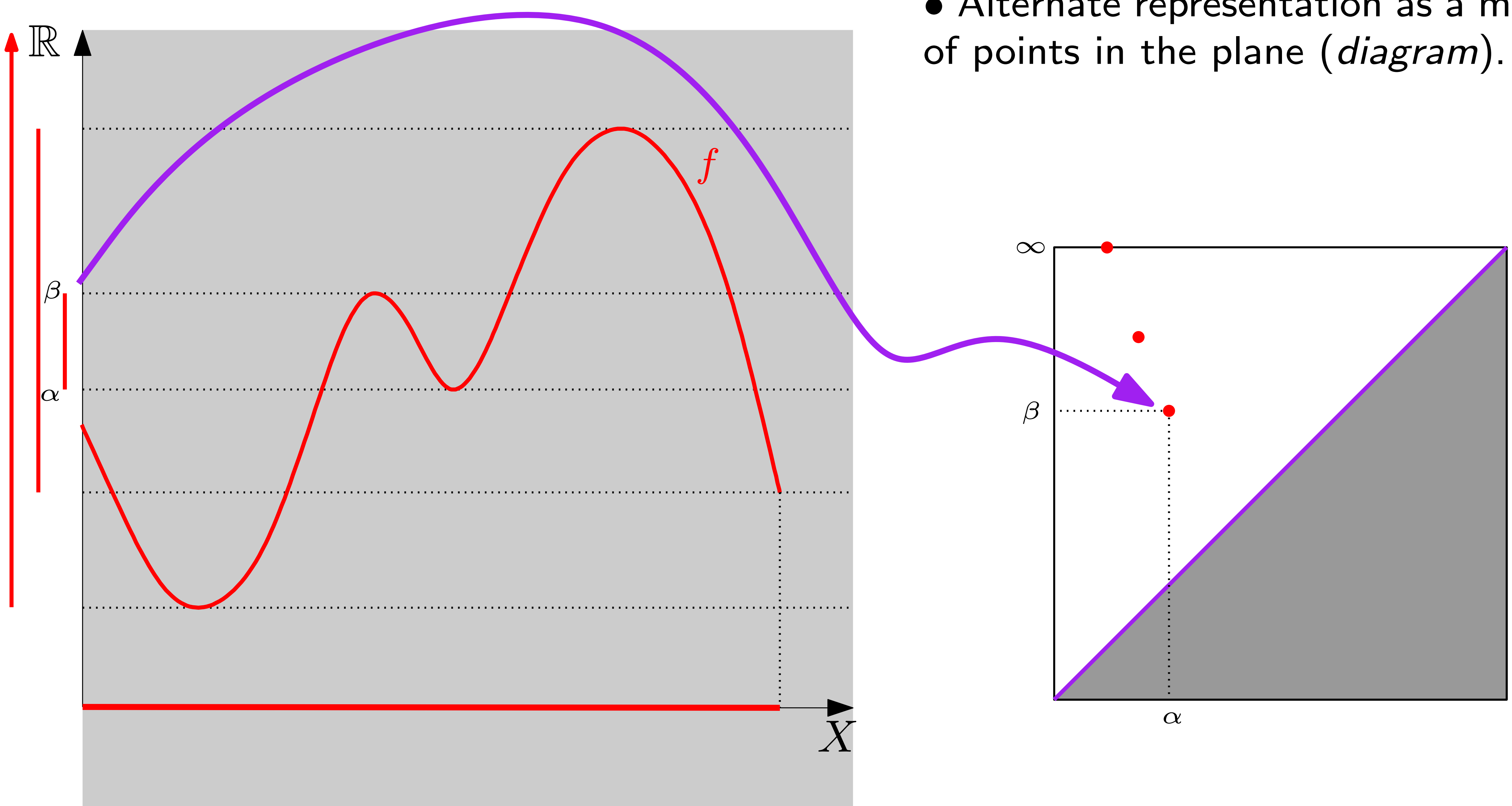


Topological Persistence (in a nutshell)

Inside the black box:

- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family
- Finite set of intervals (barcode) encodes births/deaths of topological features

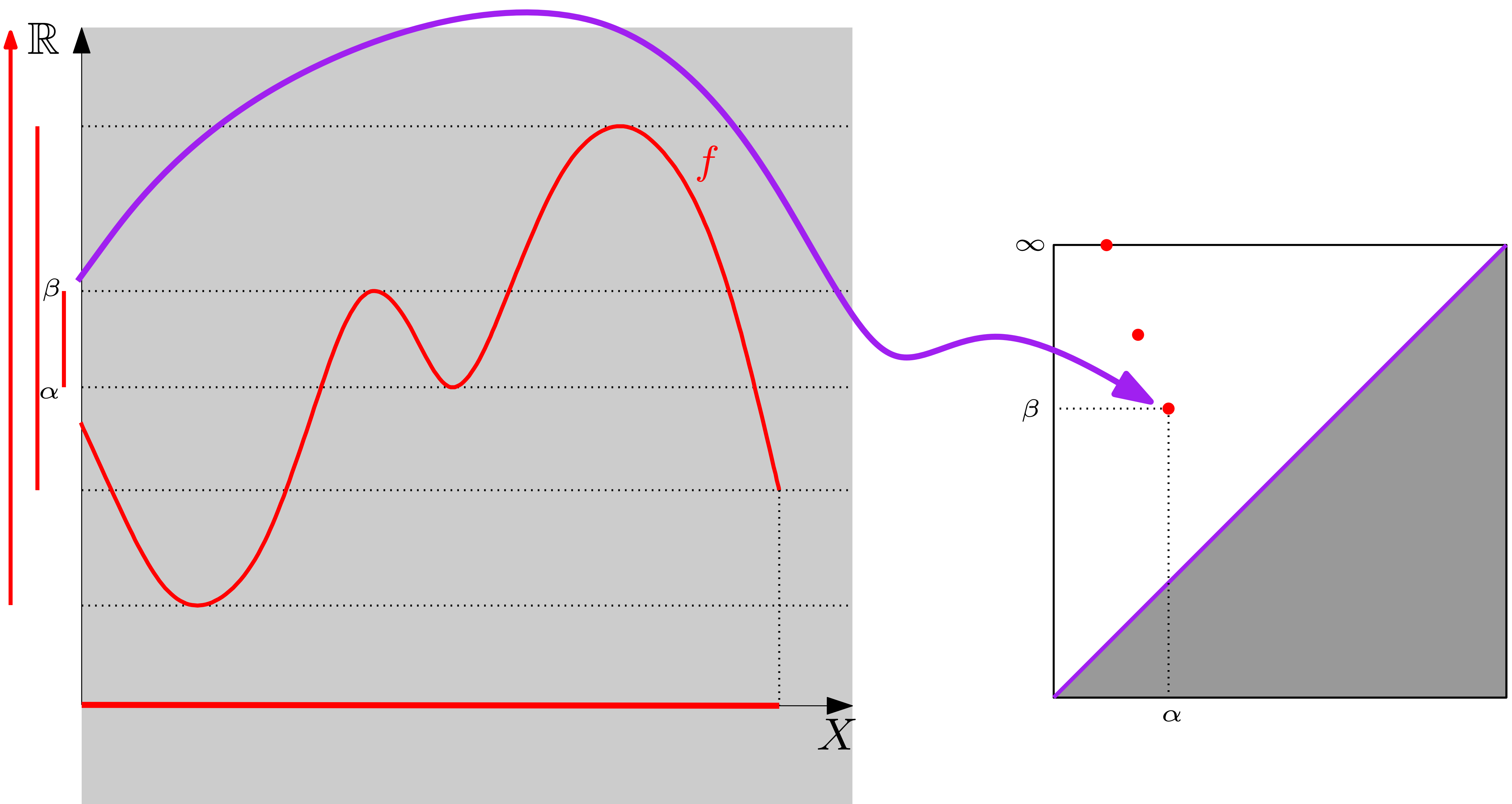
- Alternate representation as a multiset of points in the plane (*diagram*).



Topological Persistence (in a nutshell)

Algorithm:

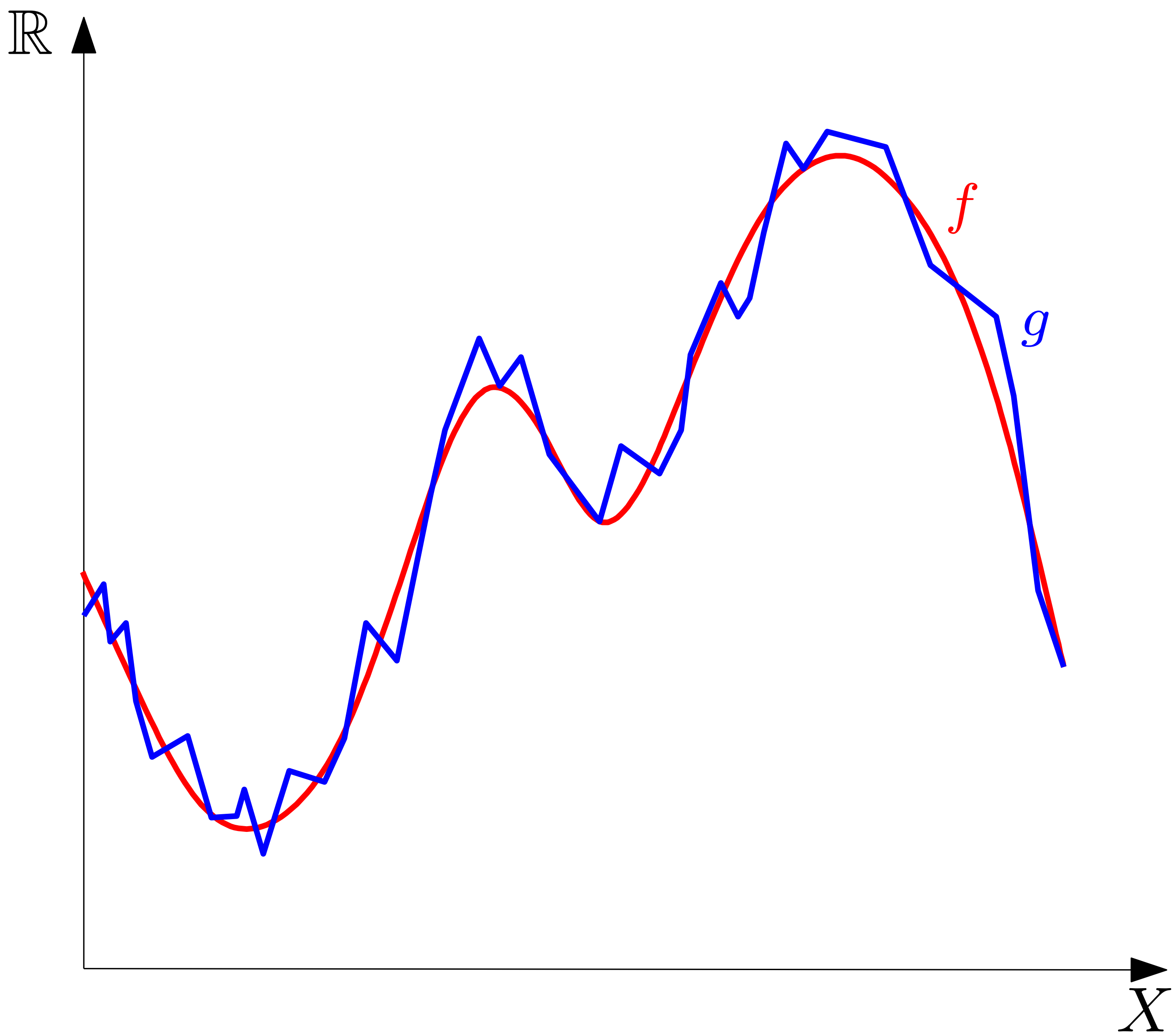
- input: graph $G = (V, E)$ + map $f : V \sqcup E \to \mathbb{R}$
- procedure: scan graph by increasing f-values, update CCs by union-find



Topological Persistence (in a nutshell)

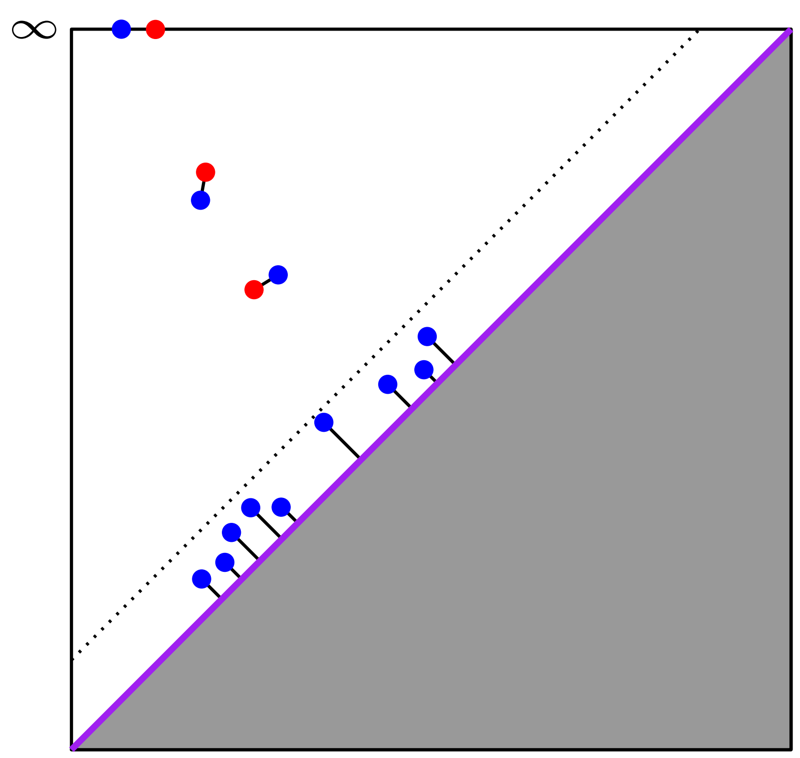
Inside the black box:

- Nested family (*filtration*) of sublevel-sets $f^{-1}((-\infty, t])$ for t ranging from $-\infty$ to $+\infty$
- Track the evolution of the topology throughout the family
- Finite set of intervals (barcode) encodes births/deaths of topological features



- Alternate representation as a multiset of points in the plane (*diagram*).

What if f is slightly perturbed?



Topological Persistence (in a nutshell)

Theorem (Stability): [Cohen-Steiner et al. 2005, Chazal, O. et al. 2009]
For any *tame* functions $f, g : \mathbb{X} \to \mathbb{R}$, $\mathrm{d}_B^\infty(\mathrm{Dg}\, f, \mathrm{Dg}\, g) \leq \|f - g\|_\infty$.

partial matching $M : \mathrm{Dg}\, f \leftrightarrow \mathrm{Dg}\, g$

cost of a matched pair $(p, q) \in M$: $\|p - q\|_\infty$

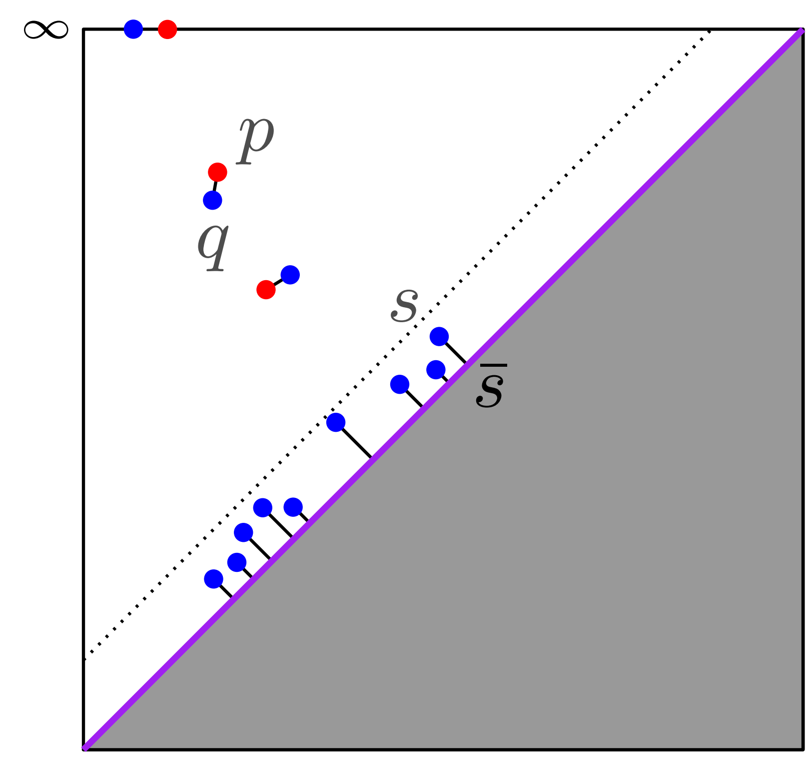
cost of an unmatched point $s \in \mathrm{Dg}\, f \sqcup \mathrm{Dg}\, g$: $\|s - \bar{s}\|_\infty$

cost of a matching:

$$\max \left\{ \sup_{(p,\,q) \text{ matched}} \|p - q\|_\infty, \sup_{s \text{ unmatched}} \|s - \bar{s}\|_\infty \right\}$$

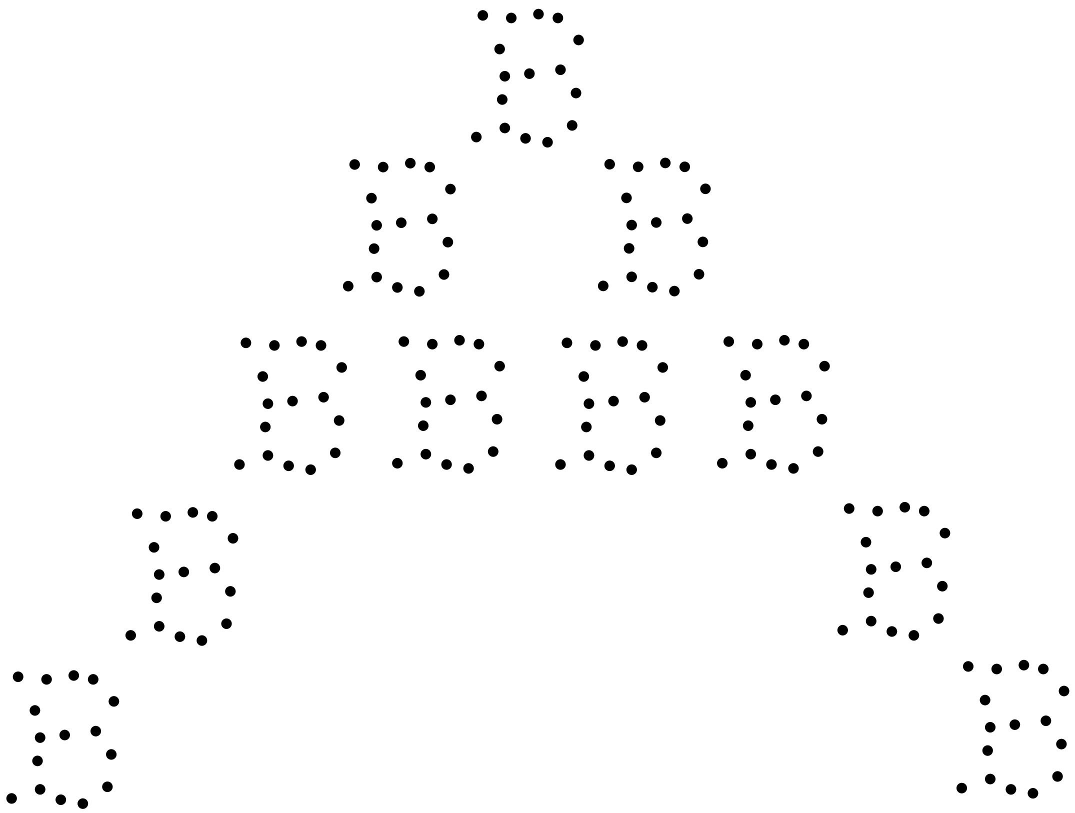
bottleneck distance:

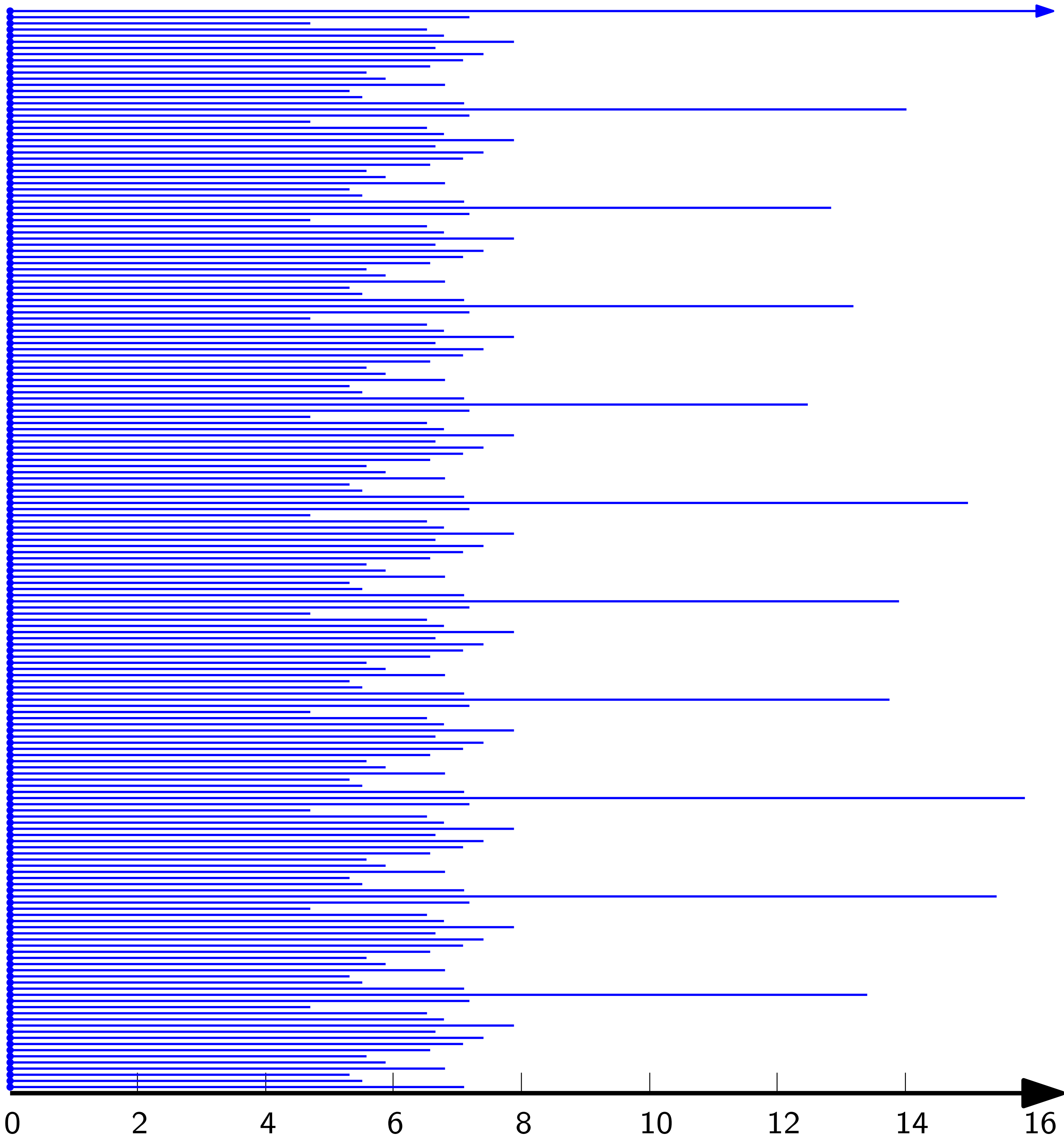
$$\mathrm{d}_B^\infty(\mathrm{Dg}\, f, \mathrm{Dg}\, g) = \inf_{M : \mathrm{Dg}\, f \leftrightarrow \mathrm{Dg}\, g} \mathrm{cost}(M)$$



Example: Distance Function

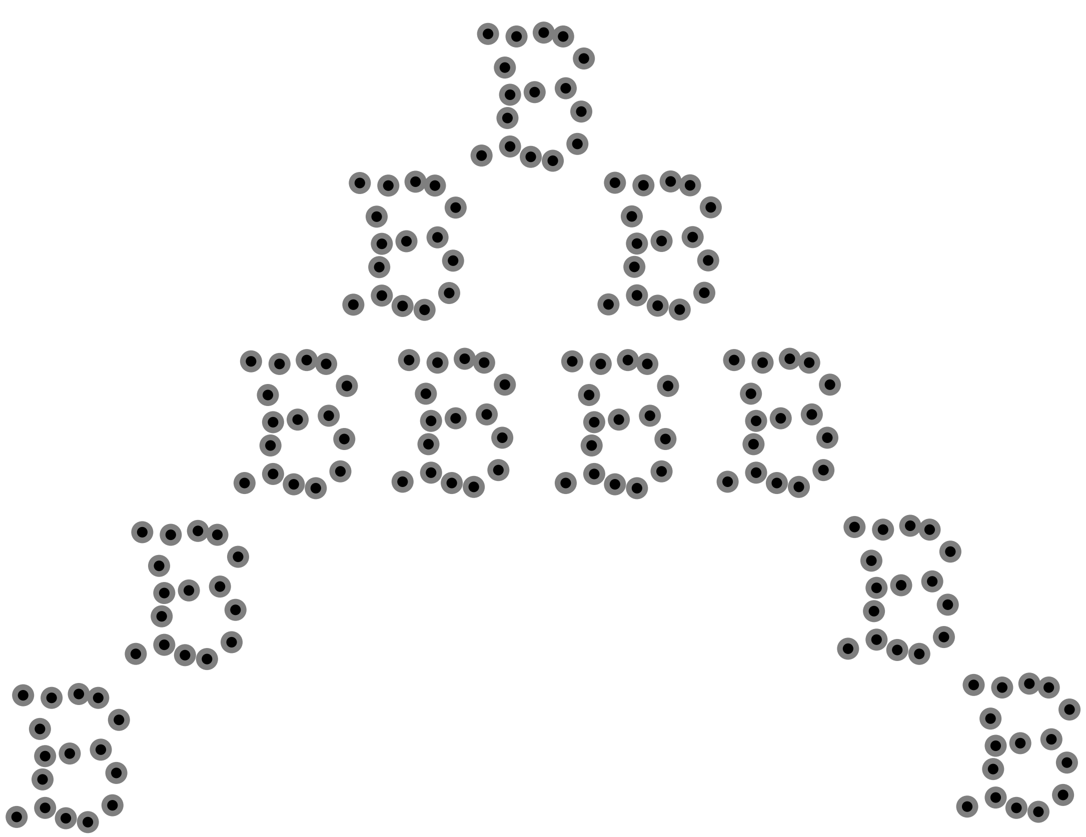
$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

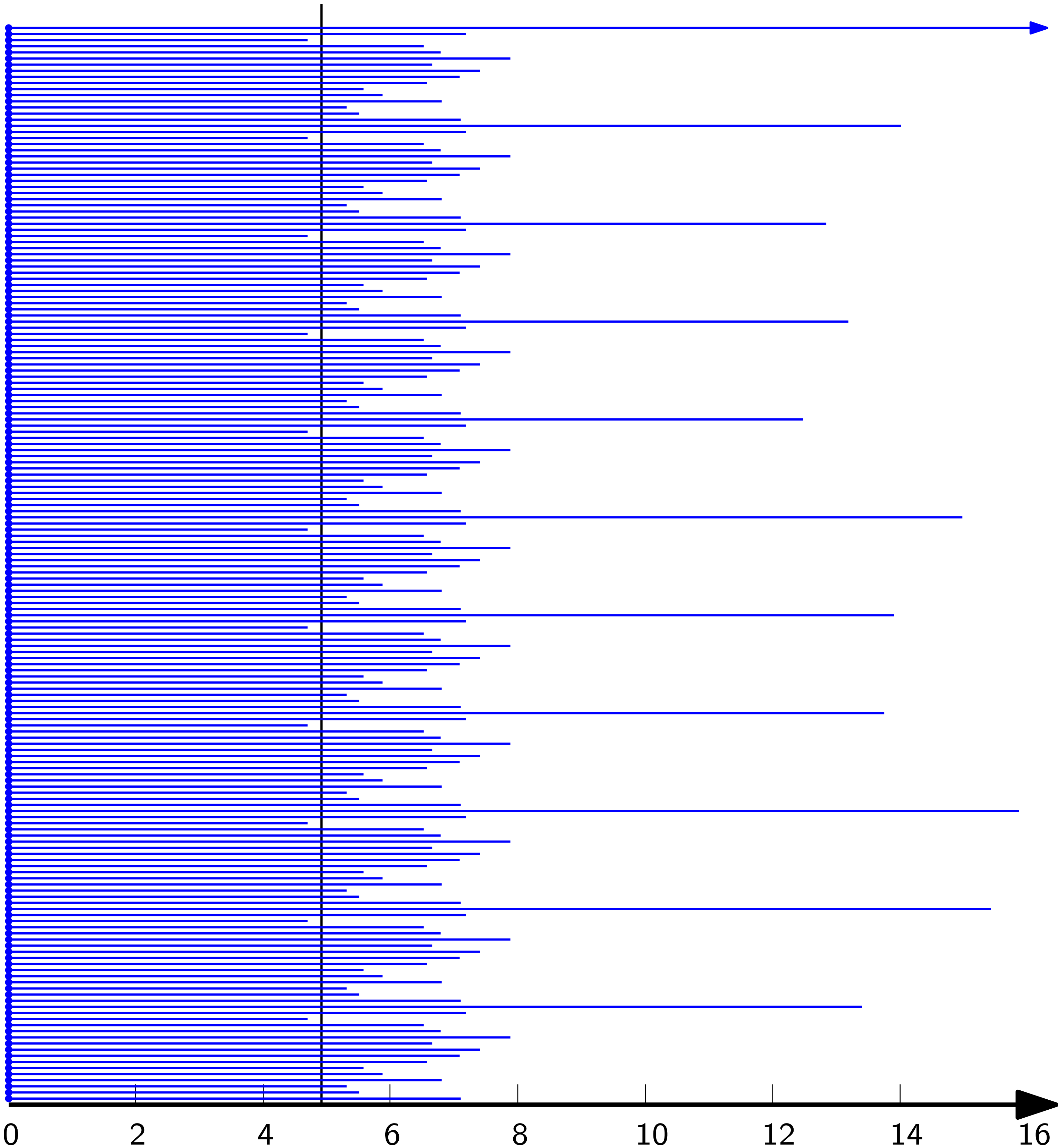




Example: Distance Function

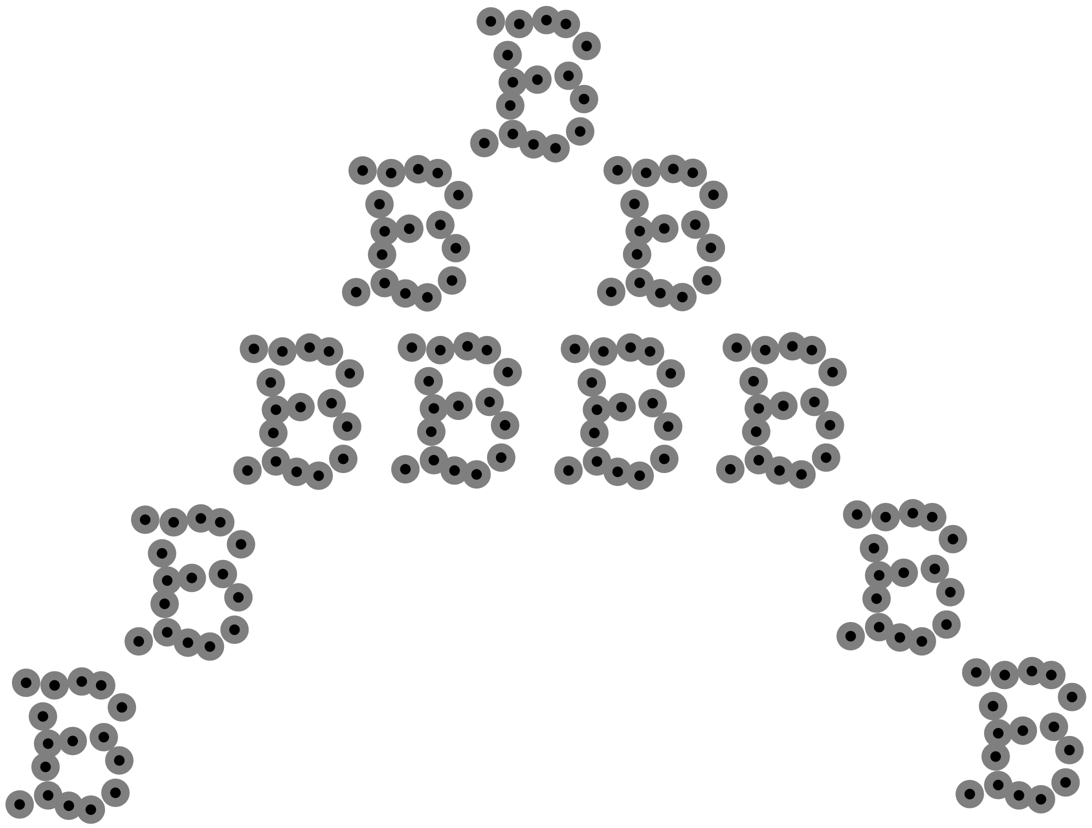
$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

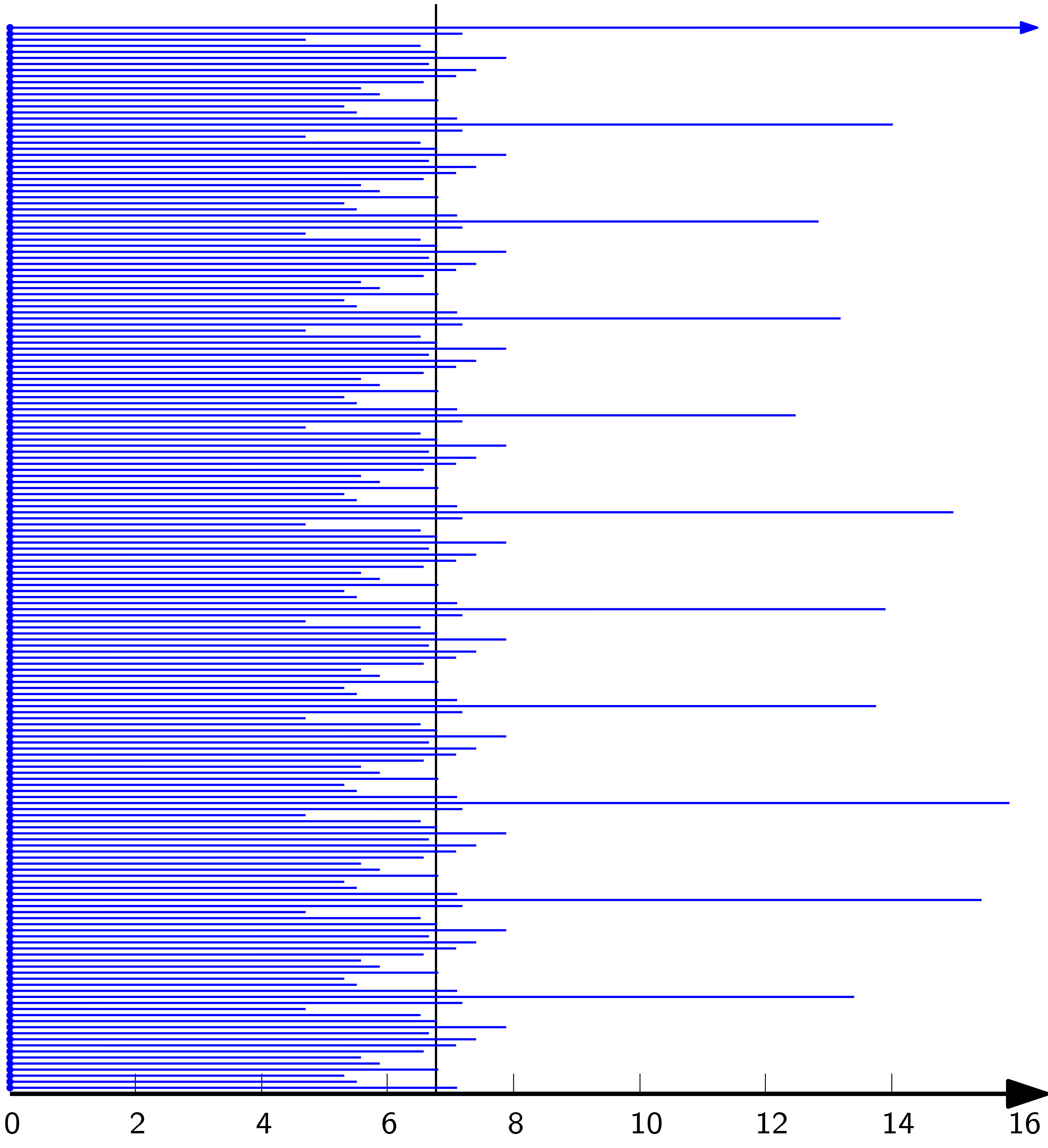




Example: Distance Function

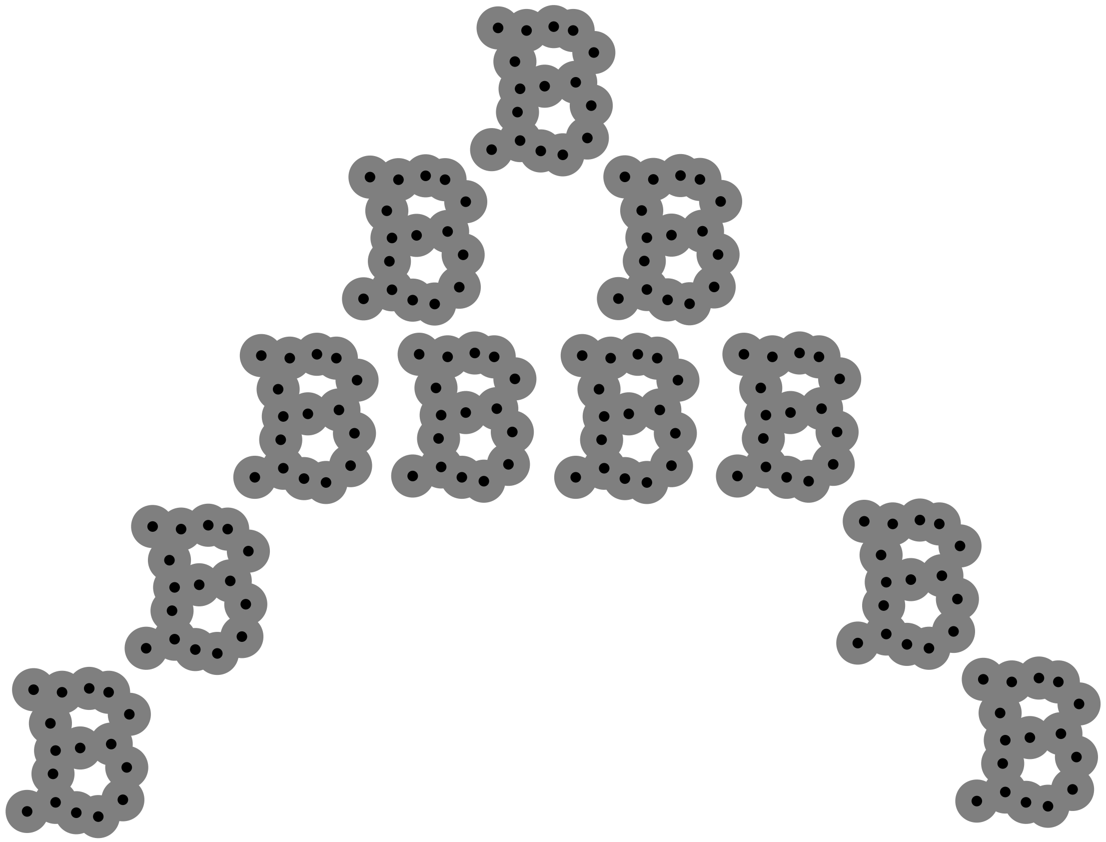
$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

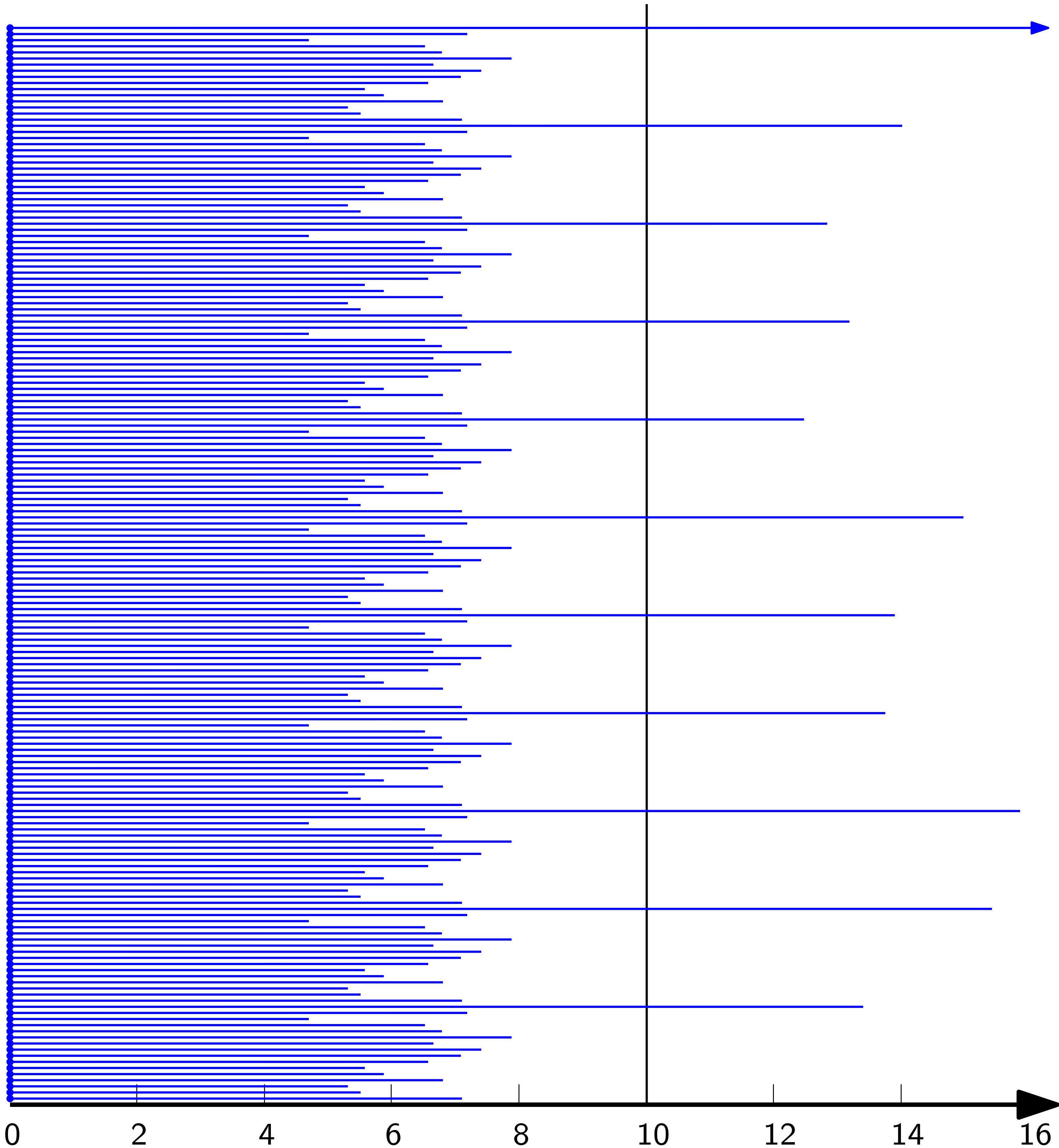




Example: Distance Function

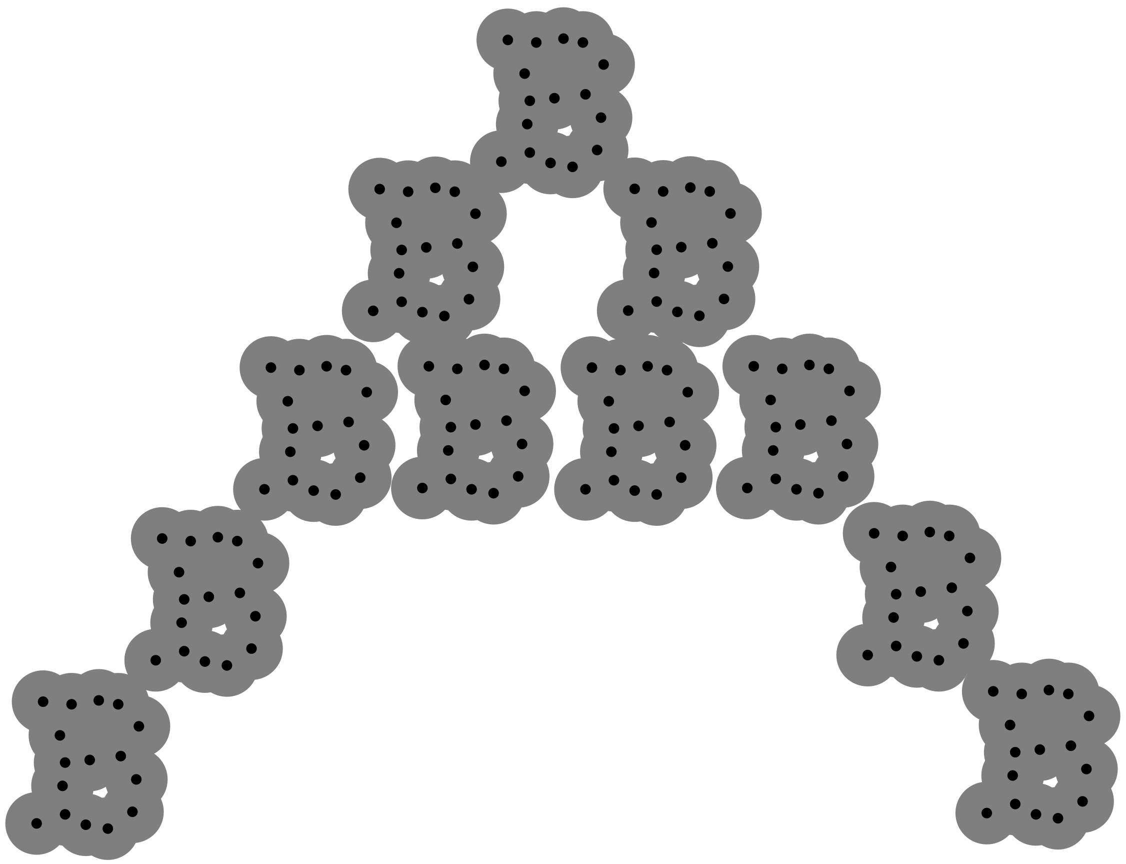
$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

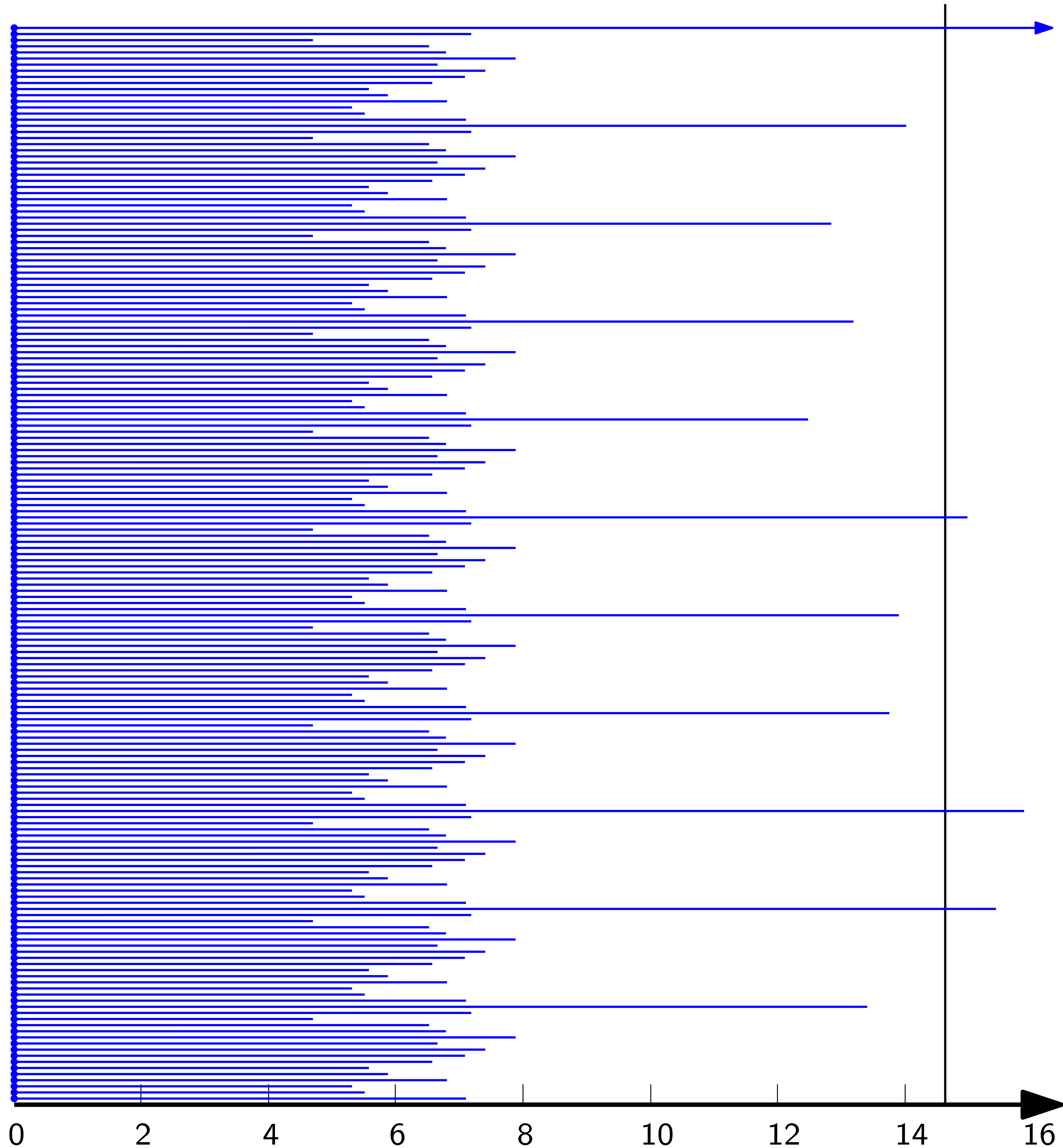




Example: Distance Function

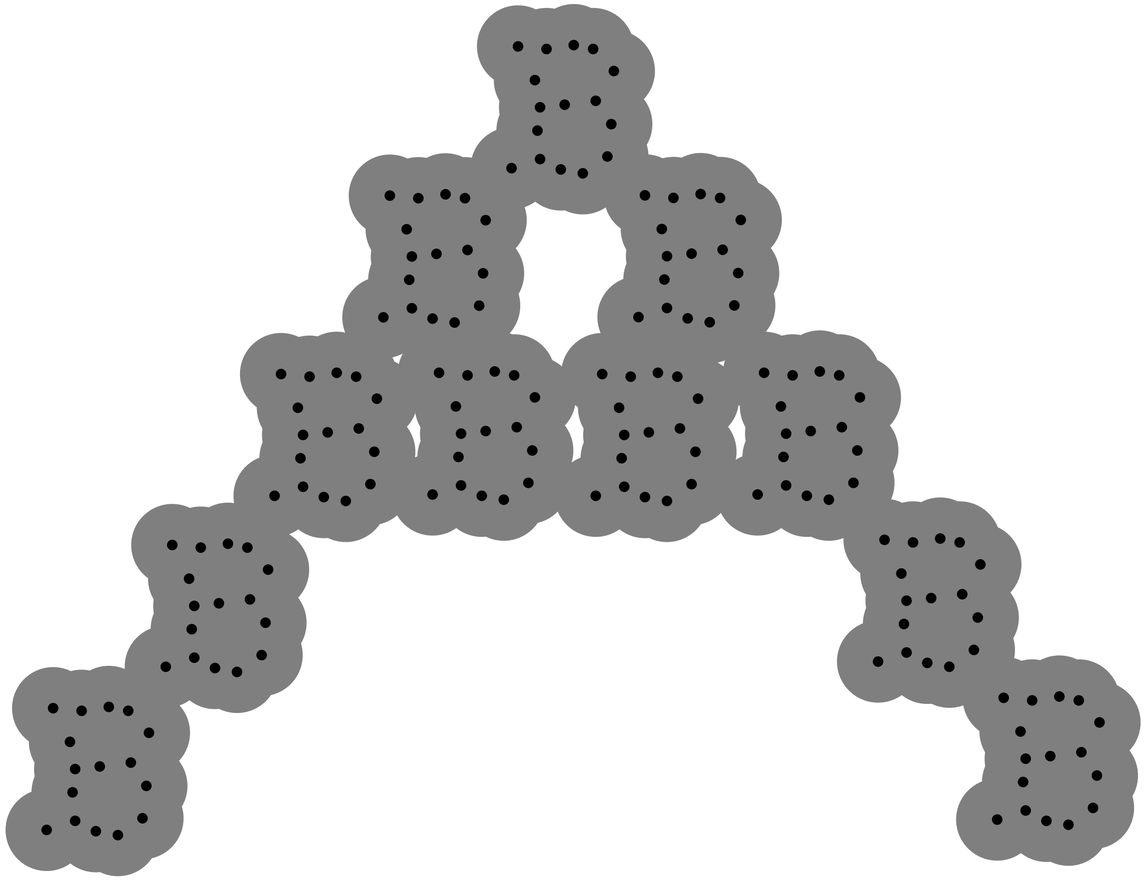
$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

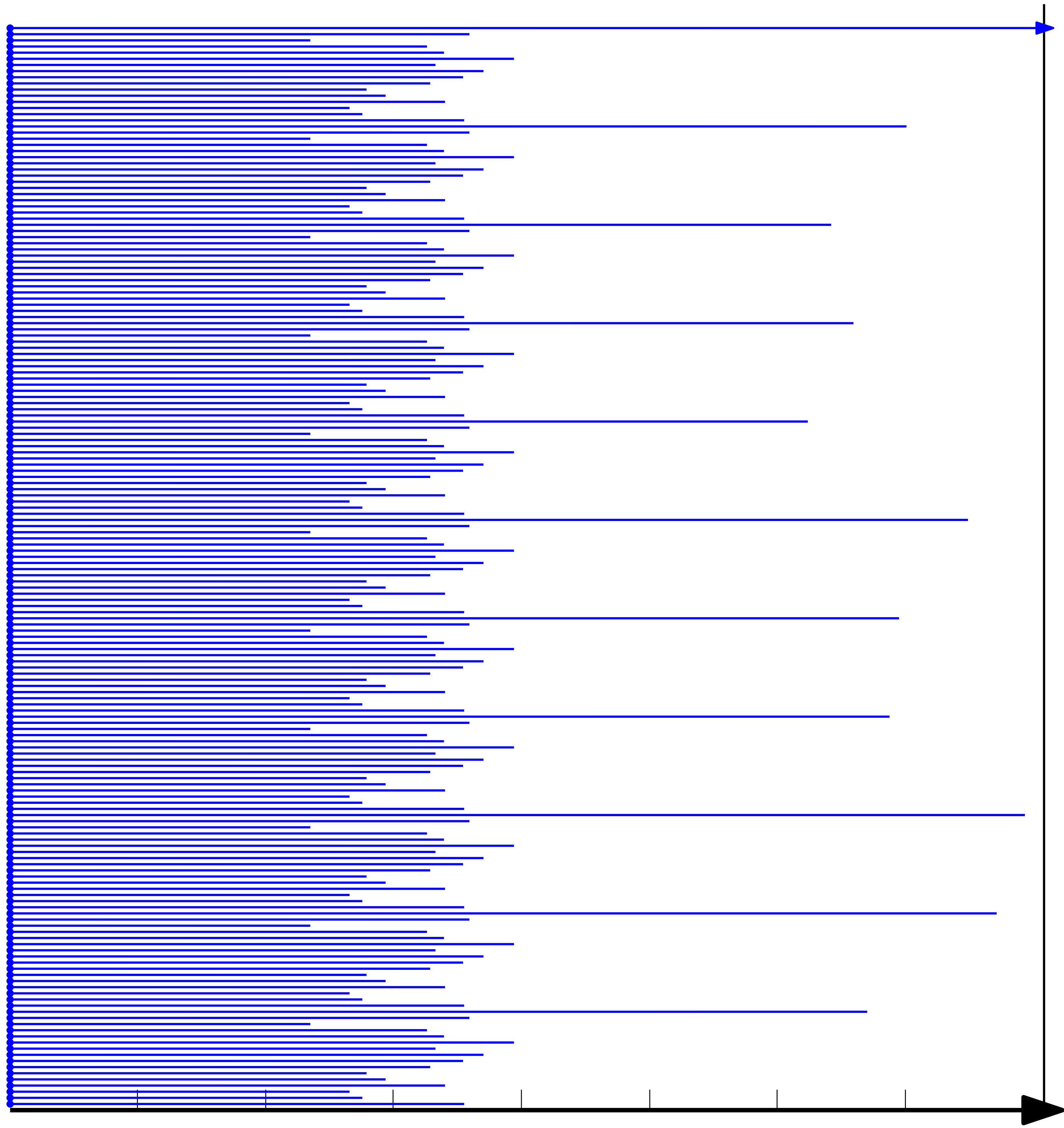




Example: Distance Function

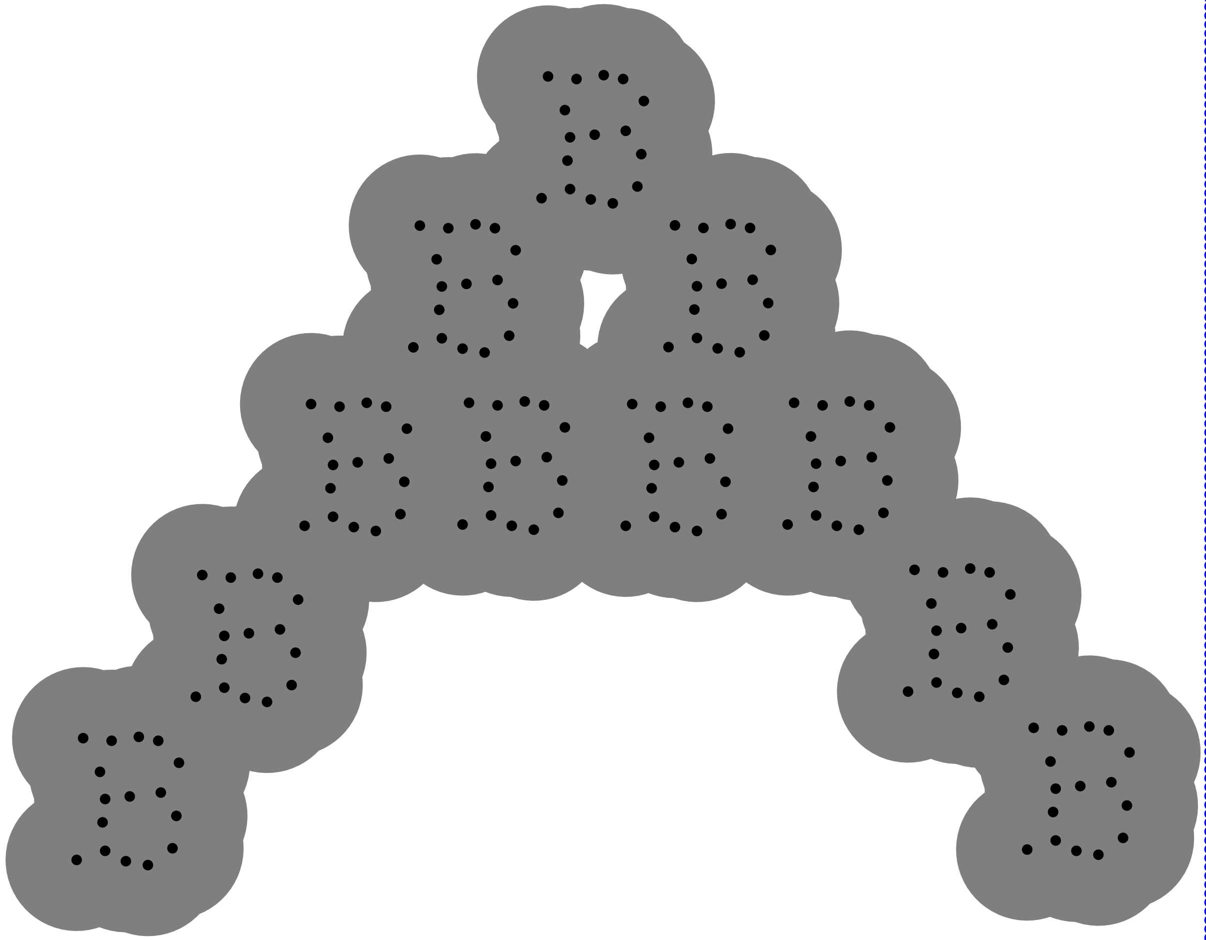
$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

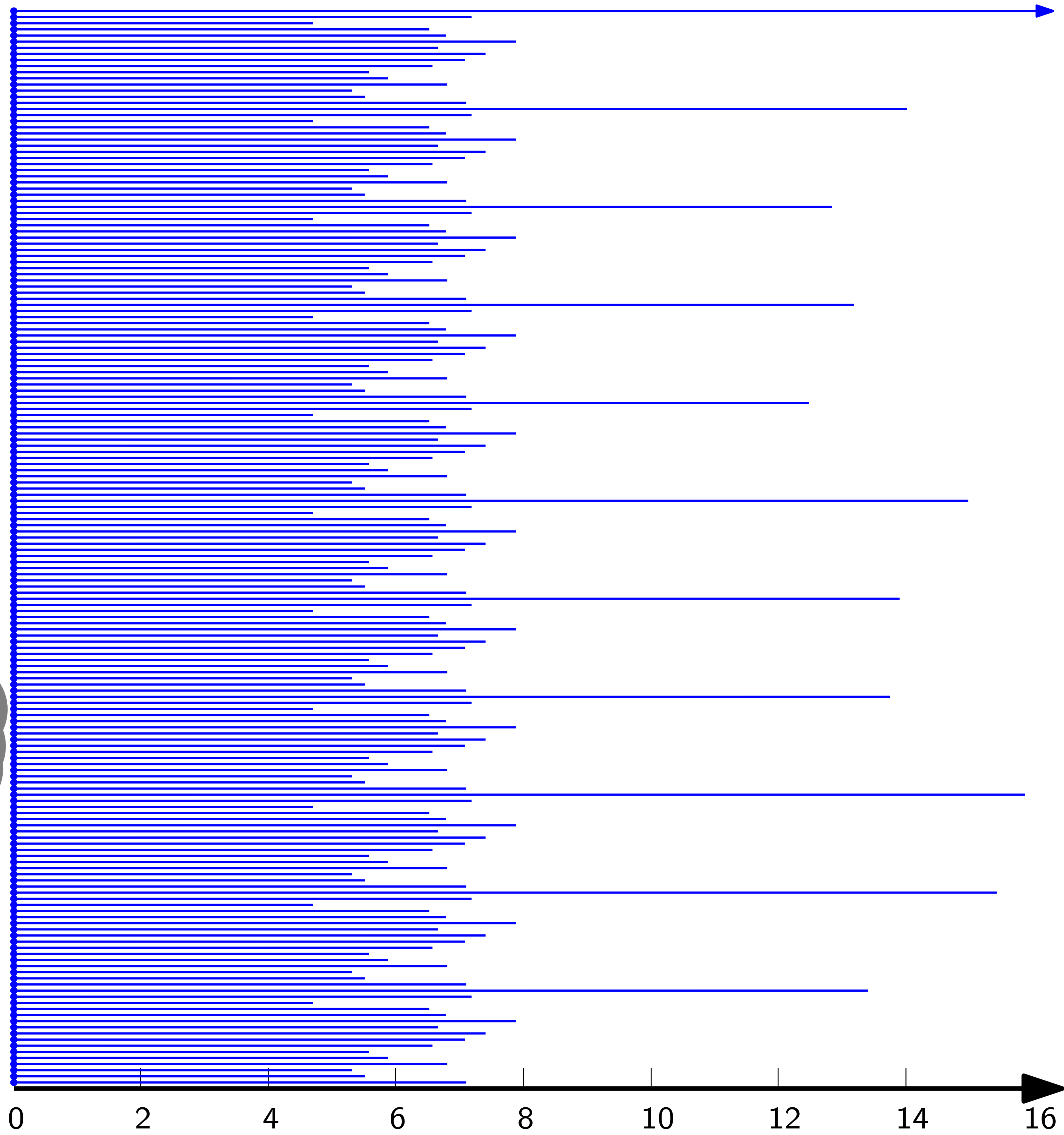




Example: Distance Function

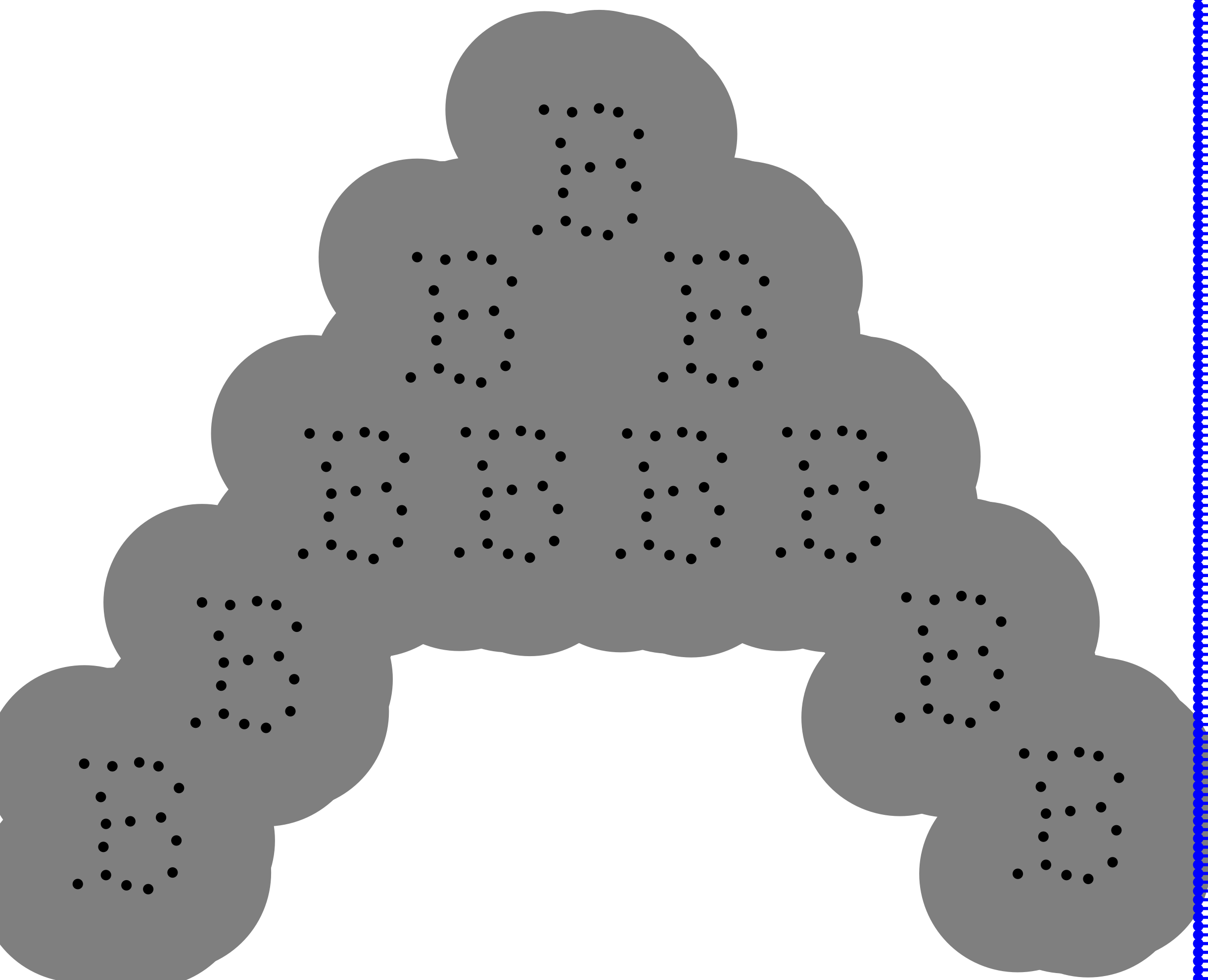
$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

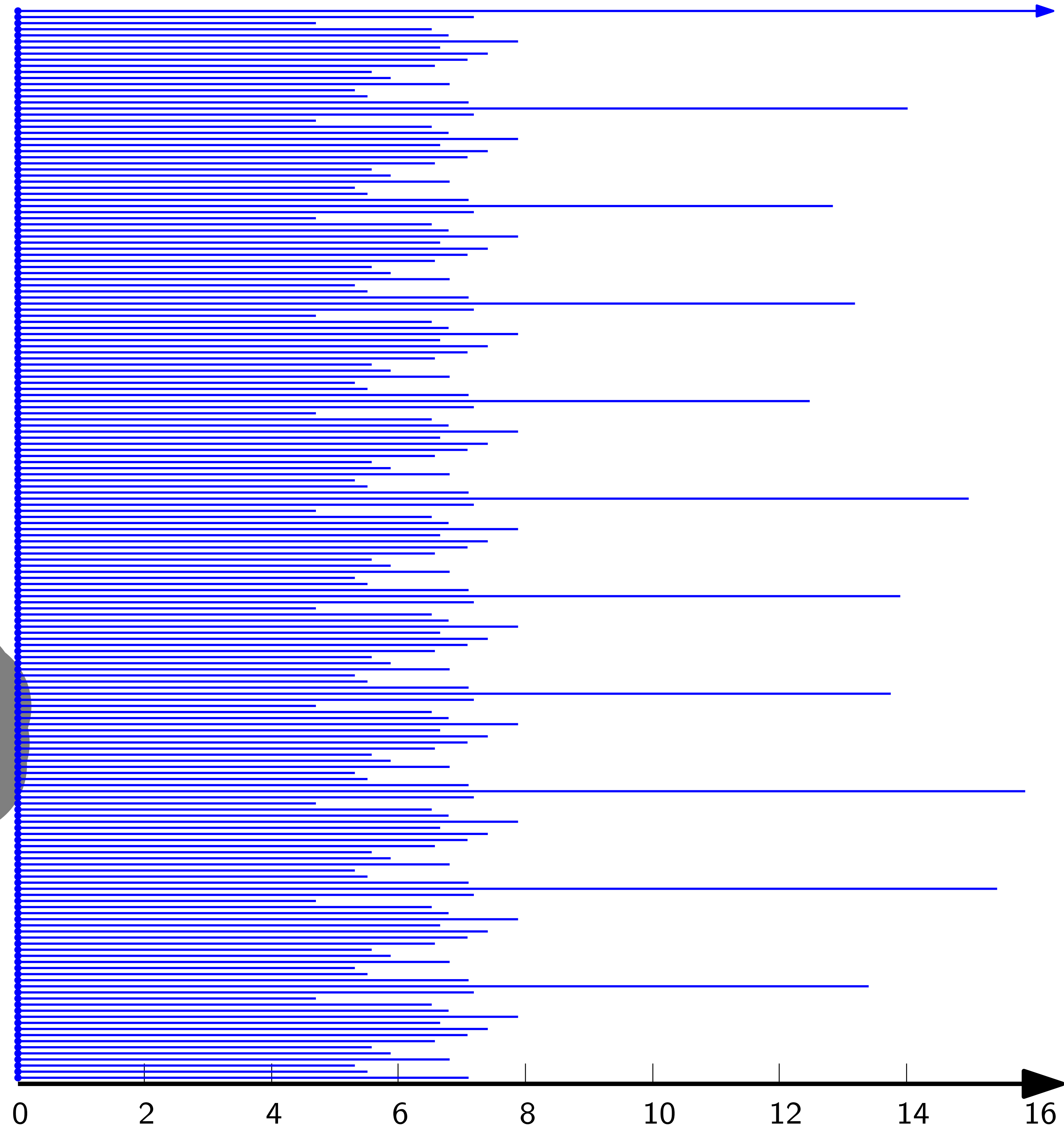




Example: Distance Function

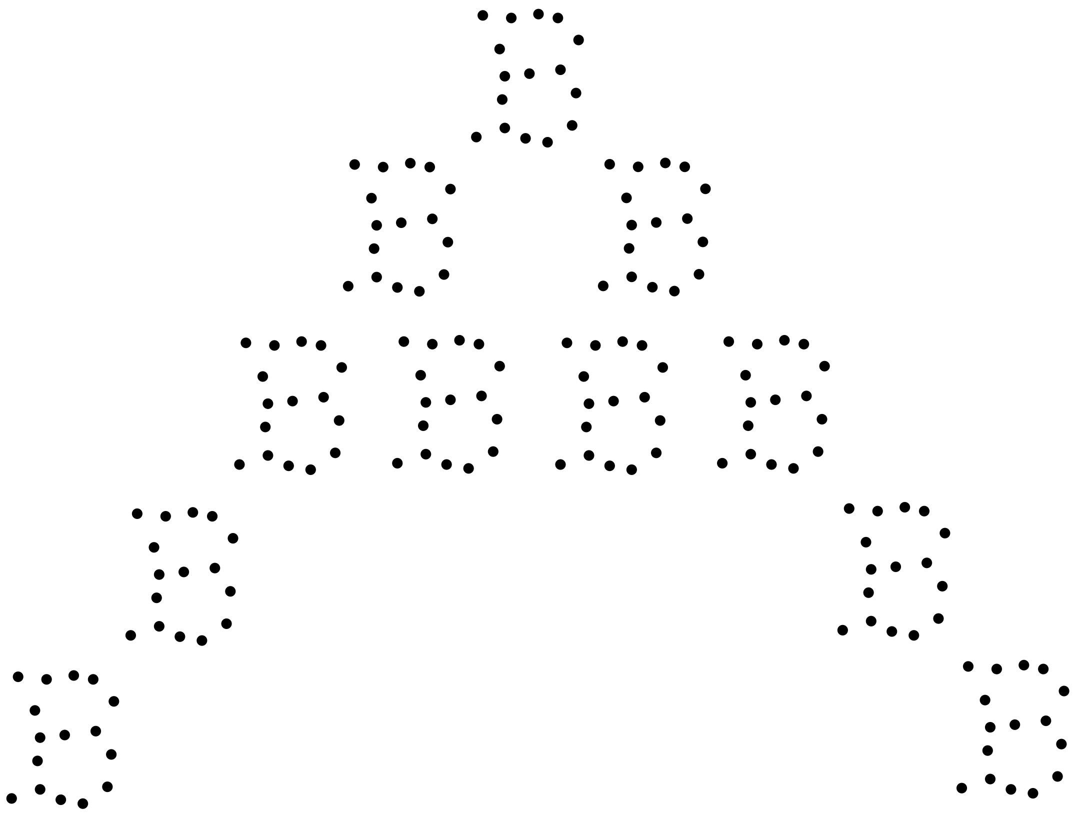
$$f_P: \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$



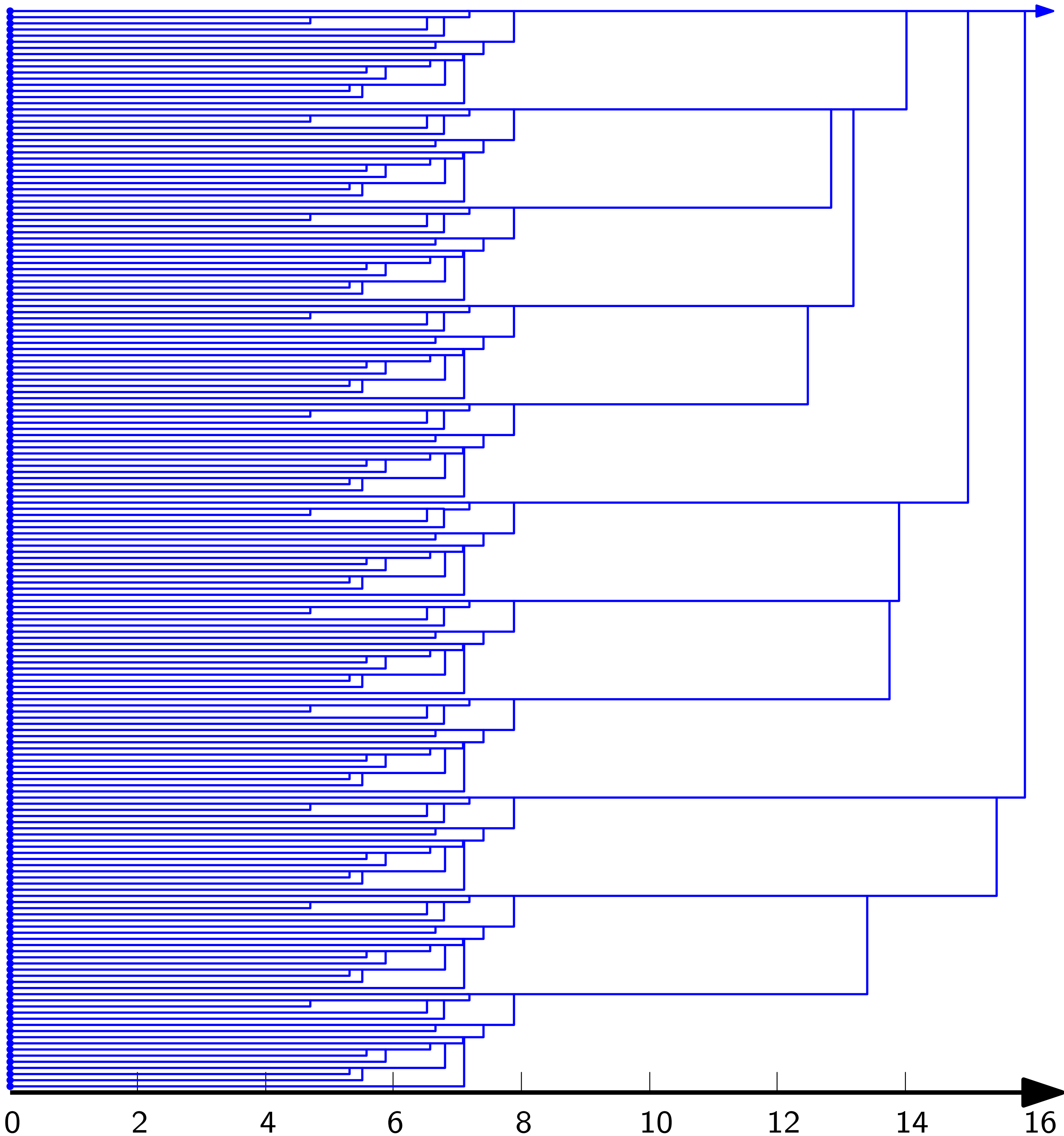


Example: Distance Function

$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$

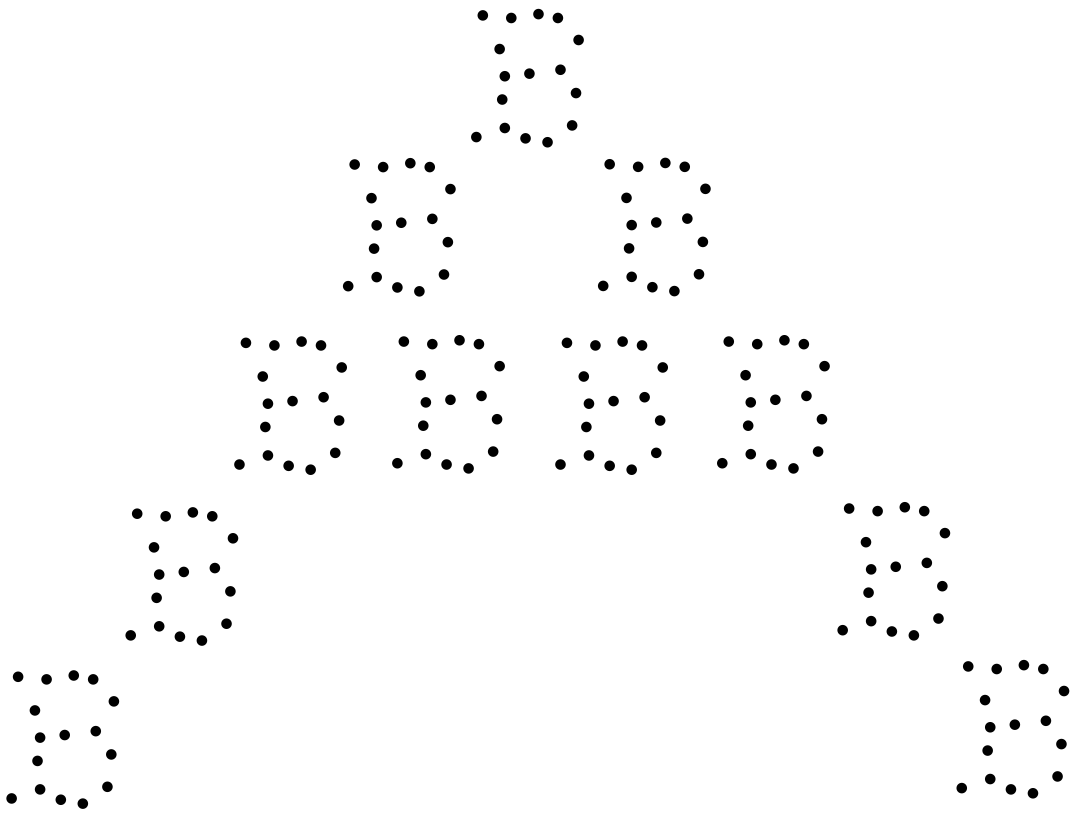


barcode $\to$ merge tree

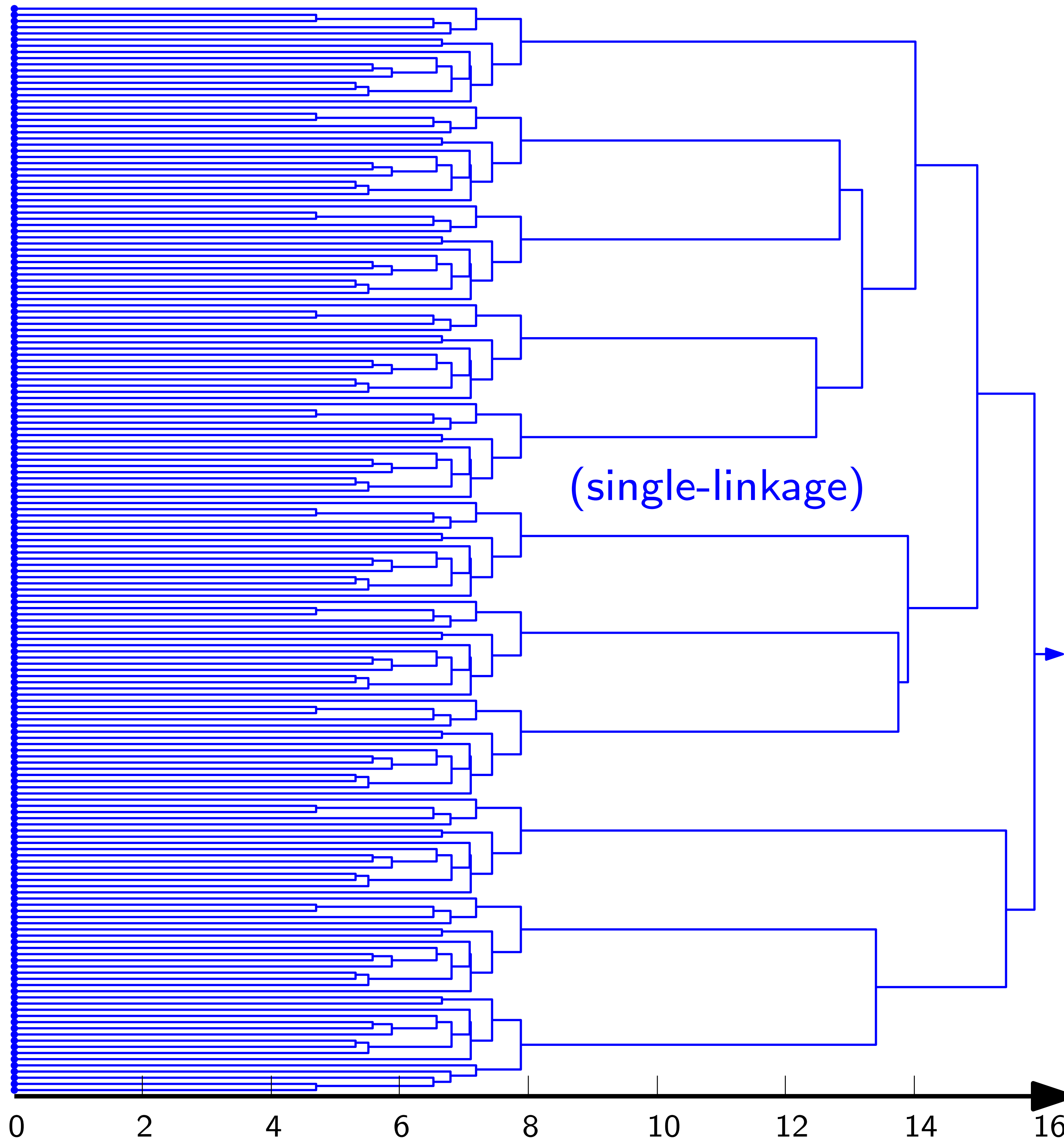


Example: Distance Function

$$f_P : \quad \mathbb{R}^2 \to \mathbb{R}$$
$$x \mapsto \min_{p \in P} \|x - p\|_2$$



barcode $\to$ merge tree $\to$ dendrogram



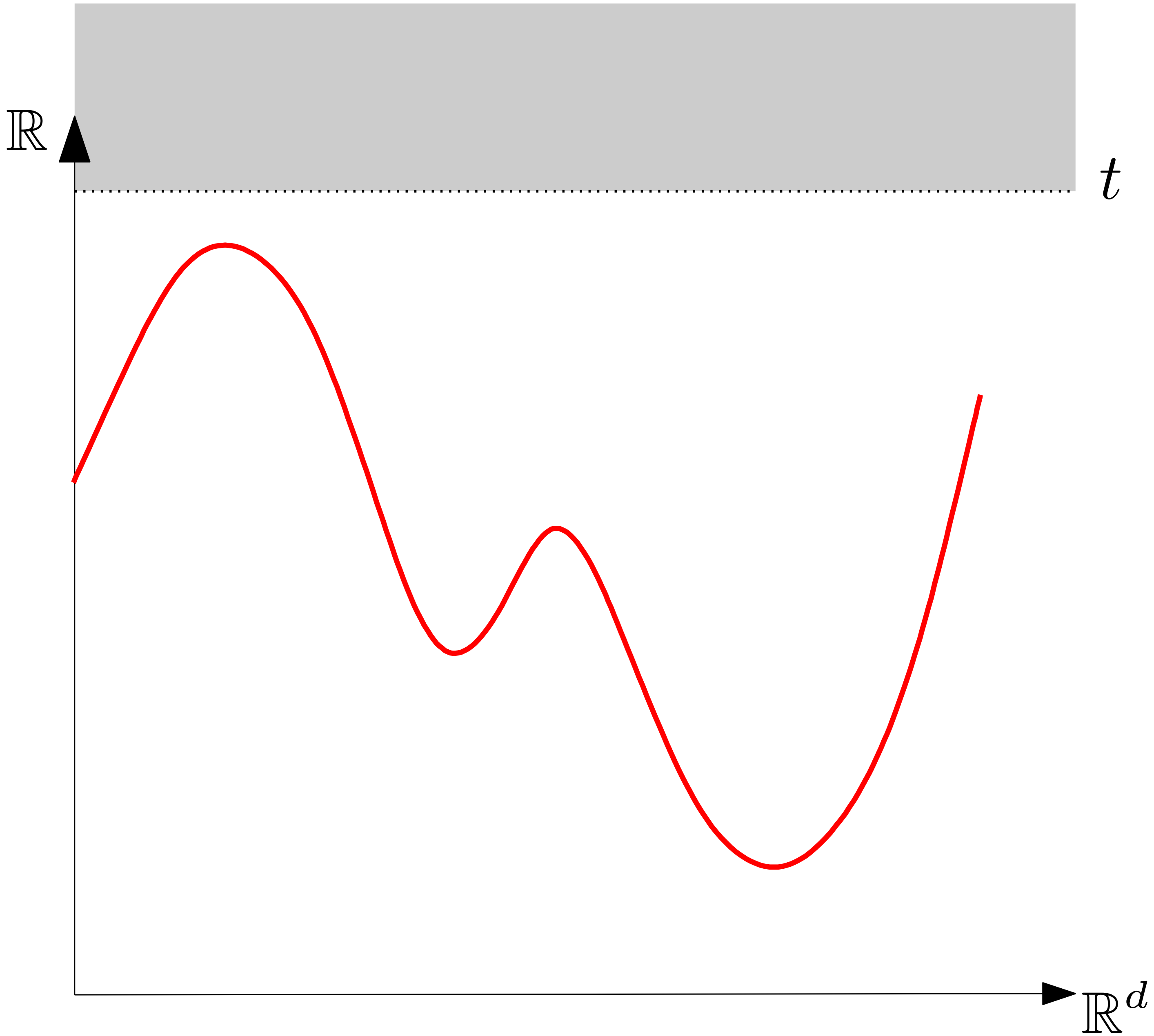
Back to Mode Seeking

(use density estimator instead of distance function)

Persistence for Mode Seeking

Given a probability density f:

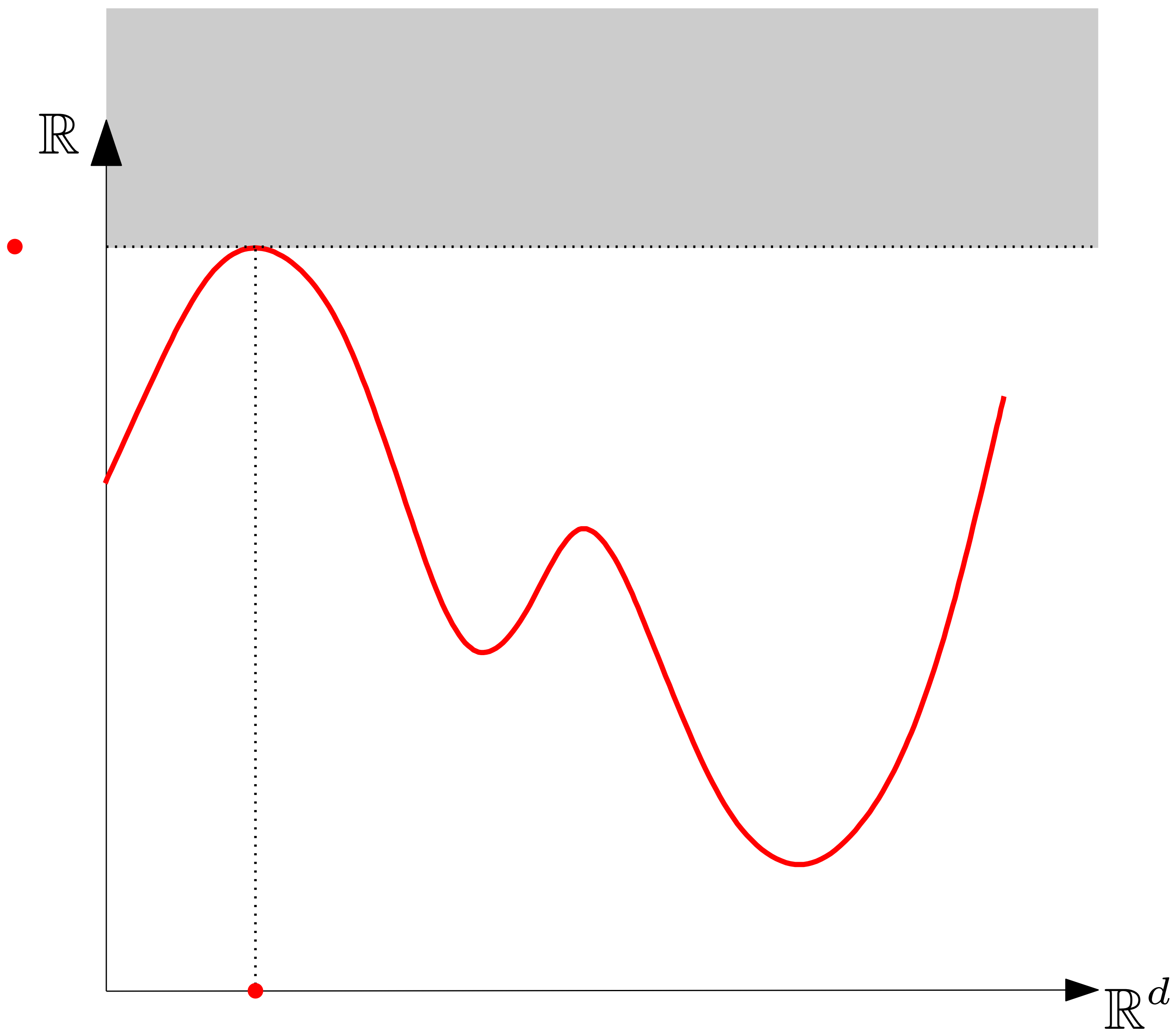
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

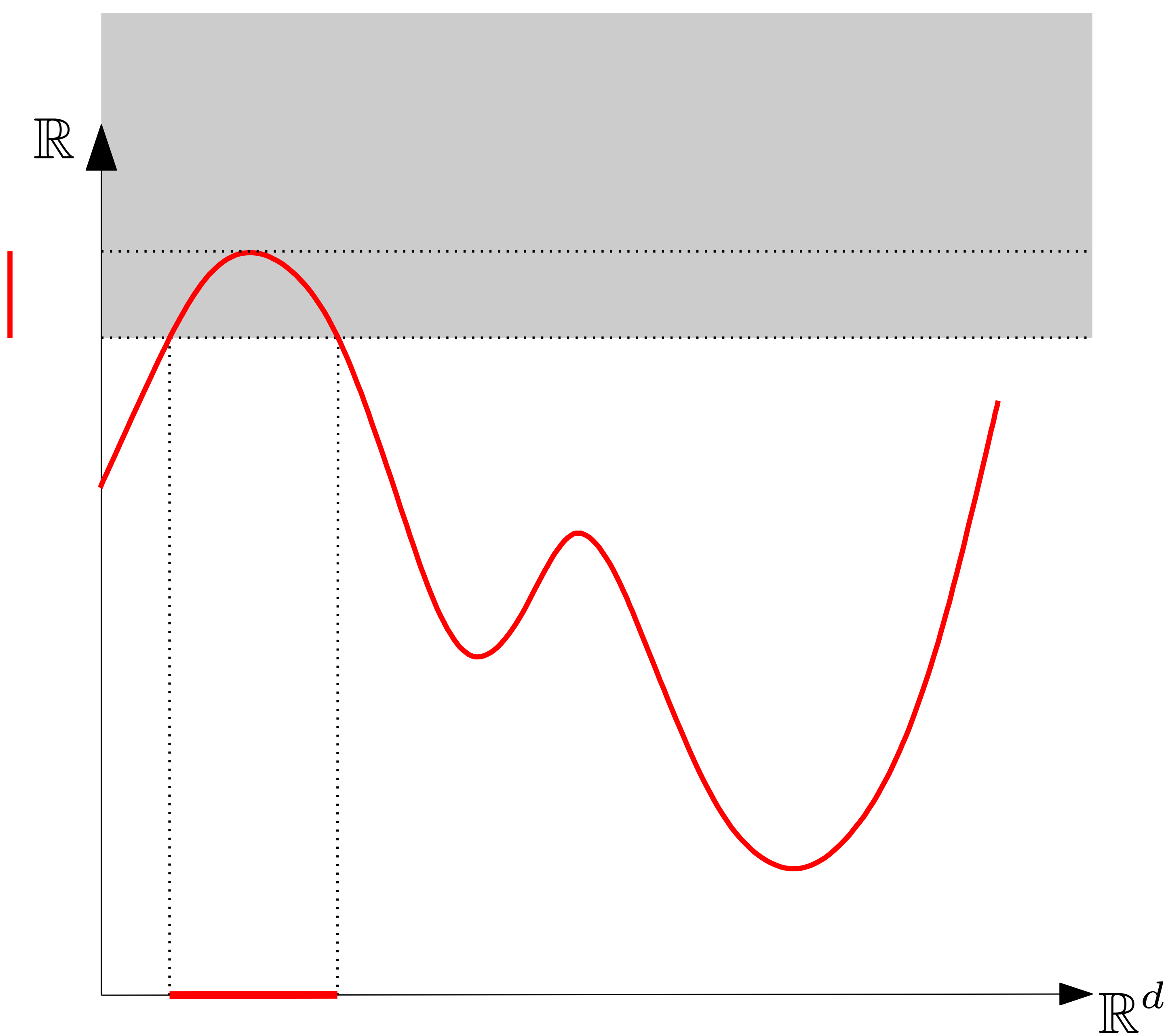
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

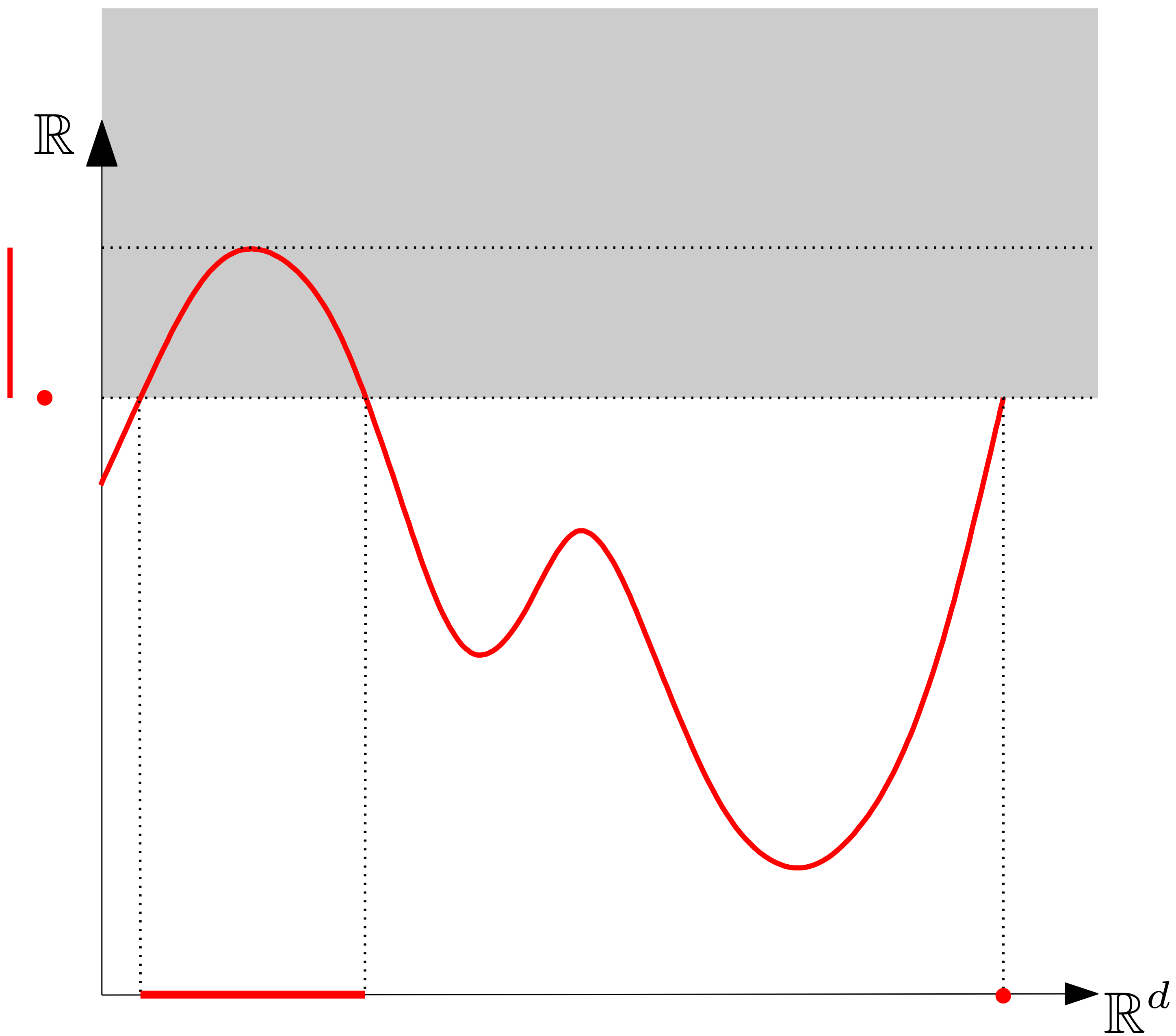
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

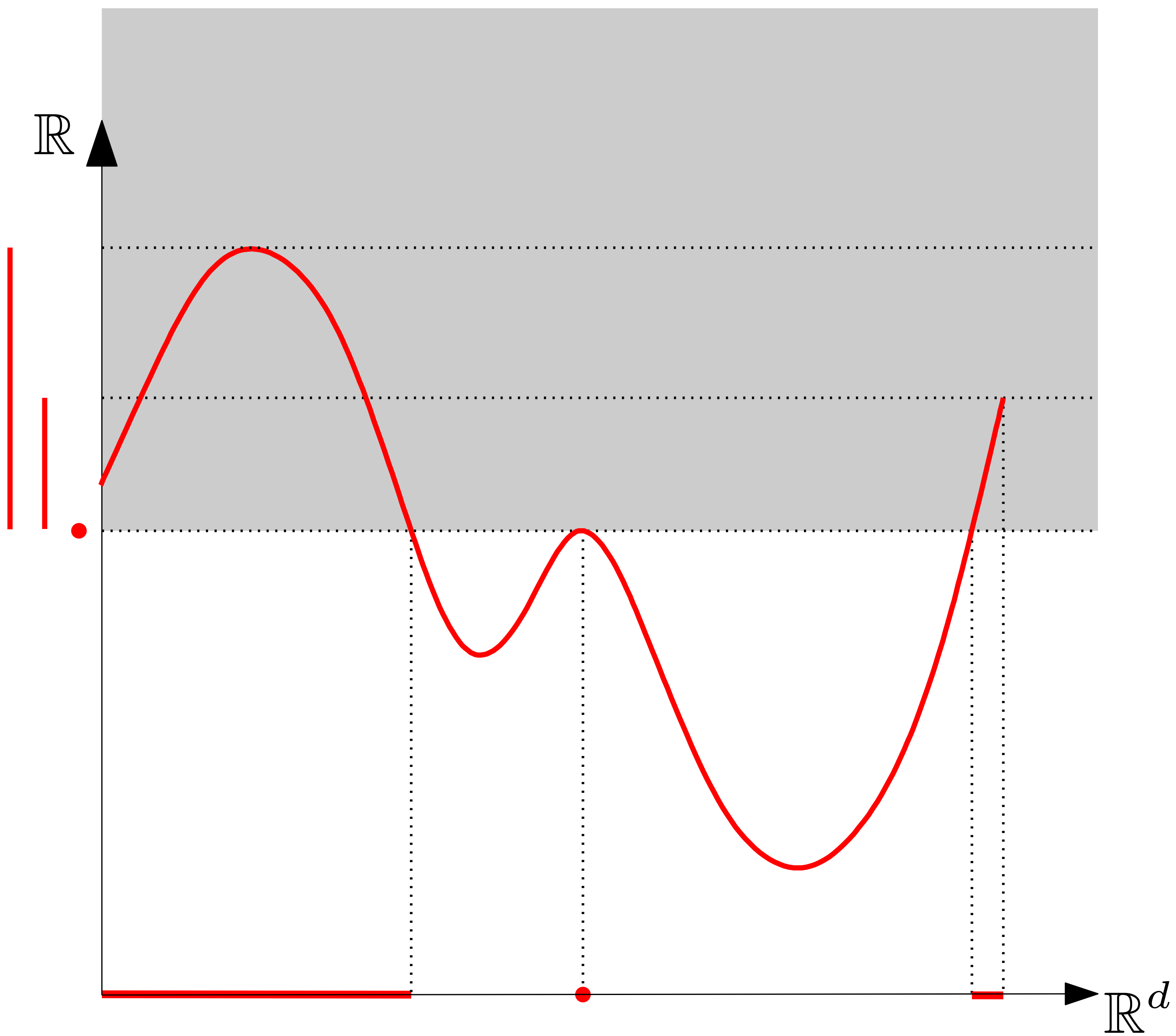
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

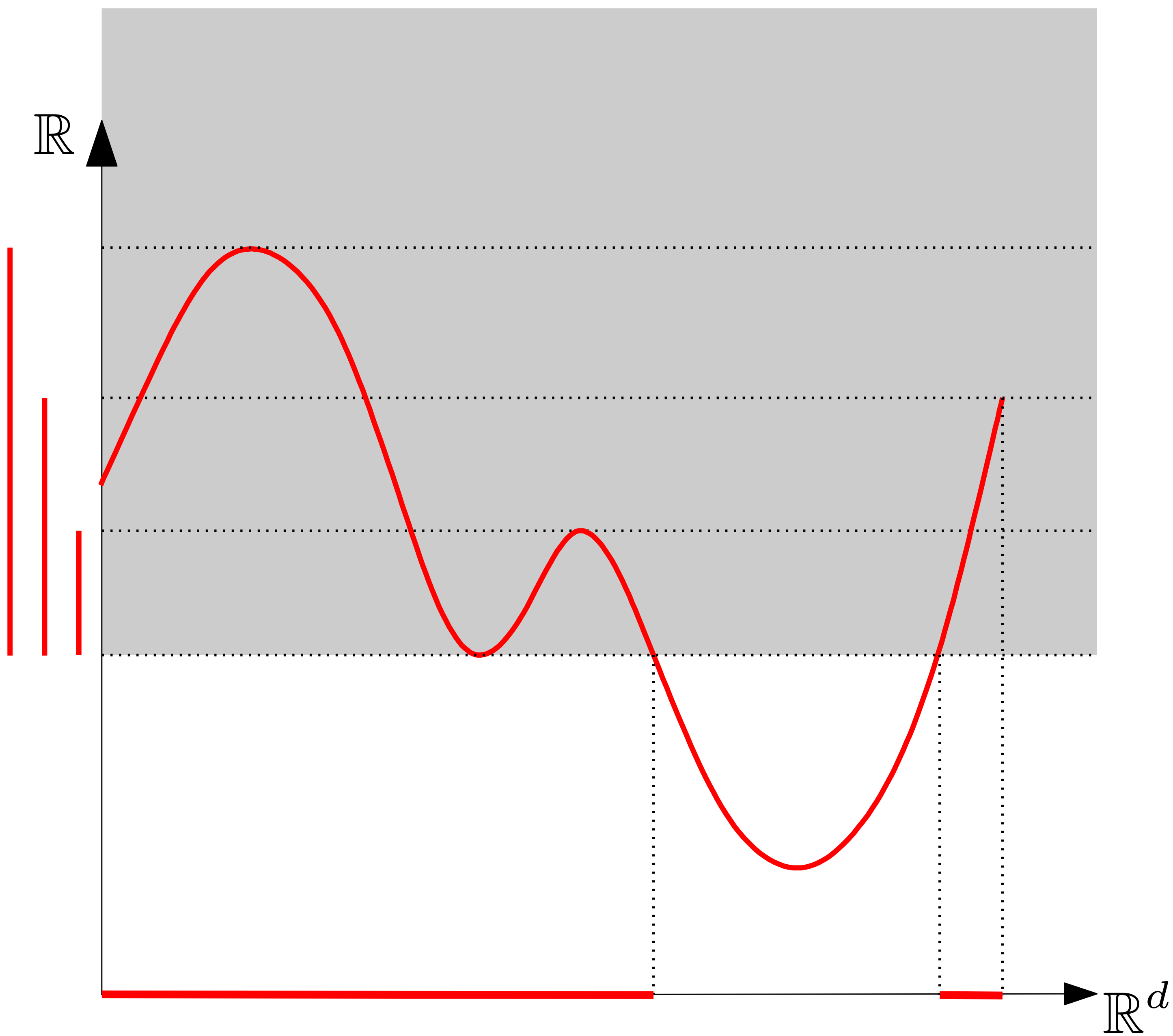
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

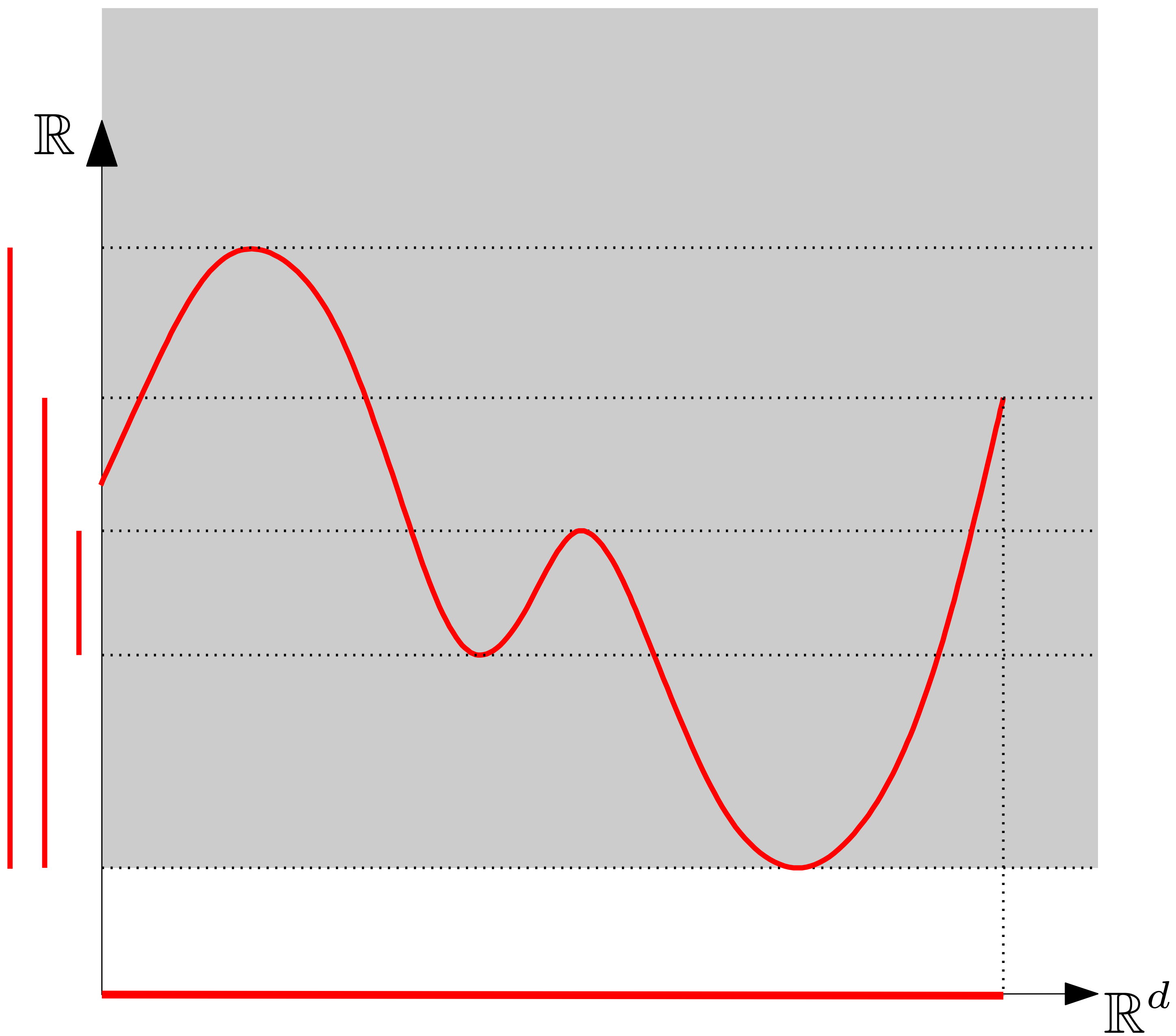
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

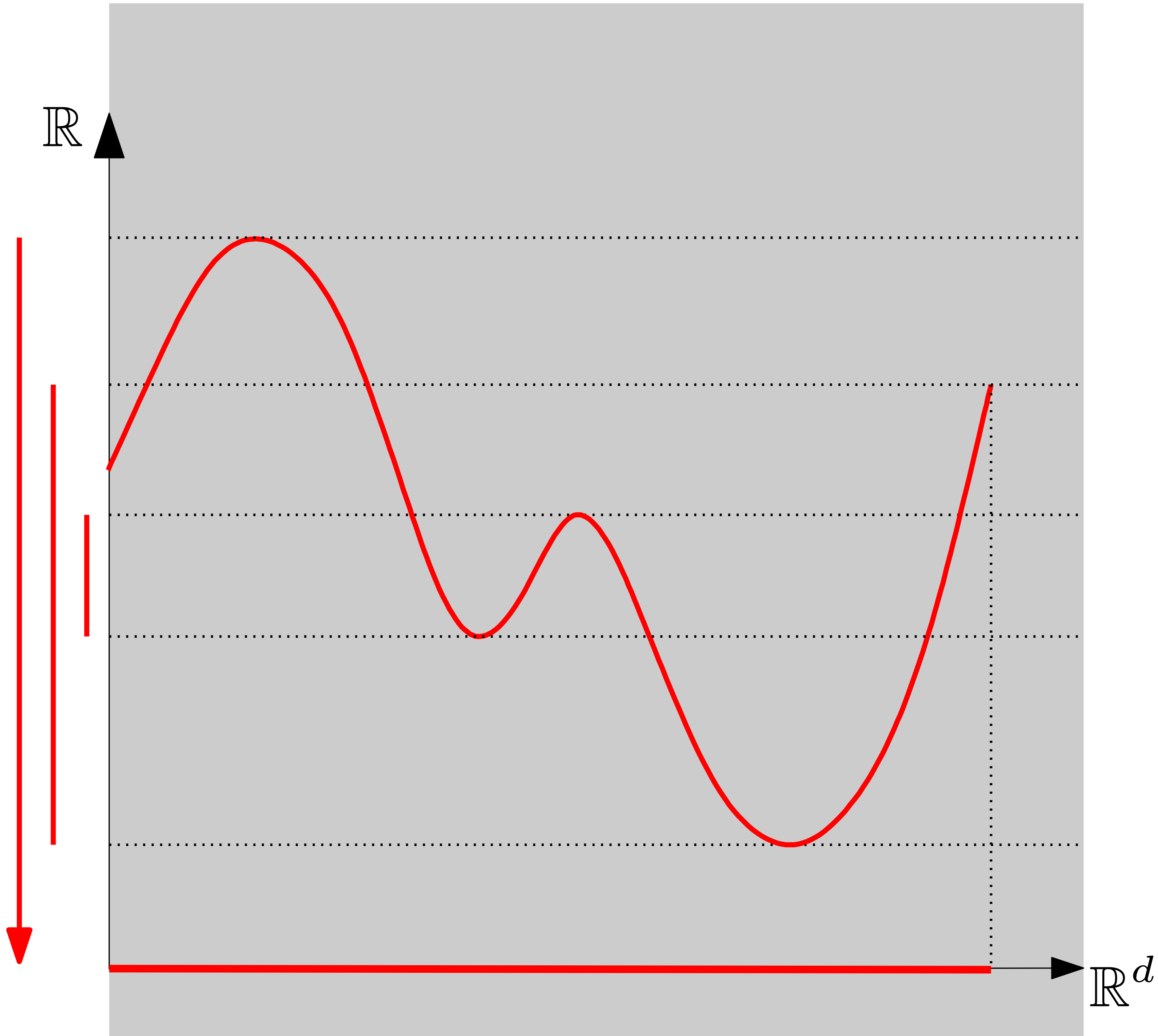
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.



Persistence for Mode Seeking

Given a probability density f:

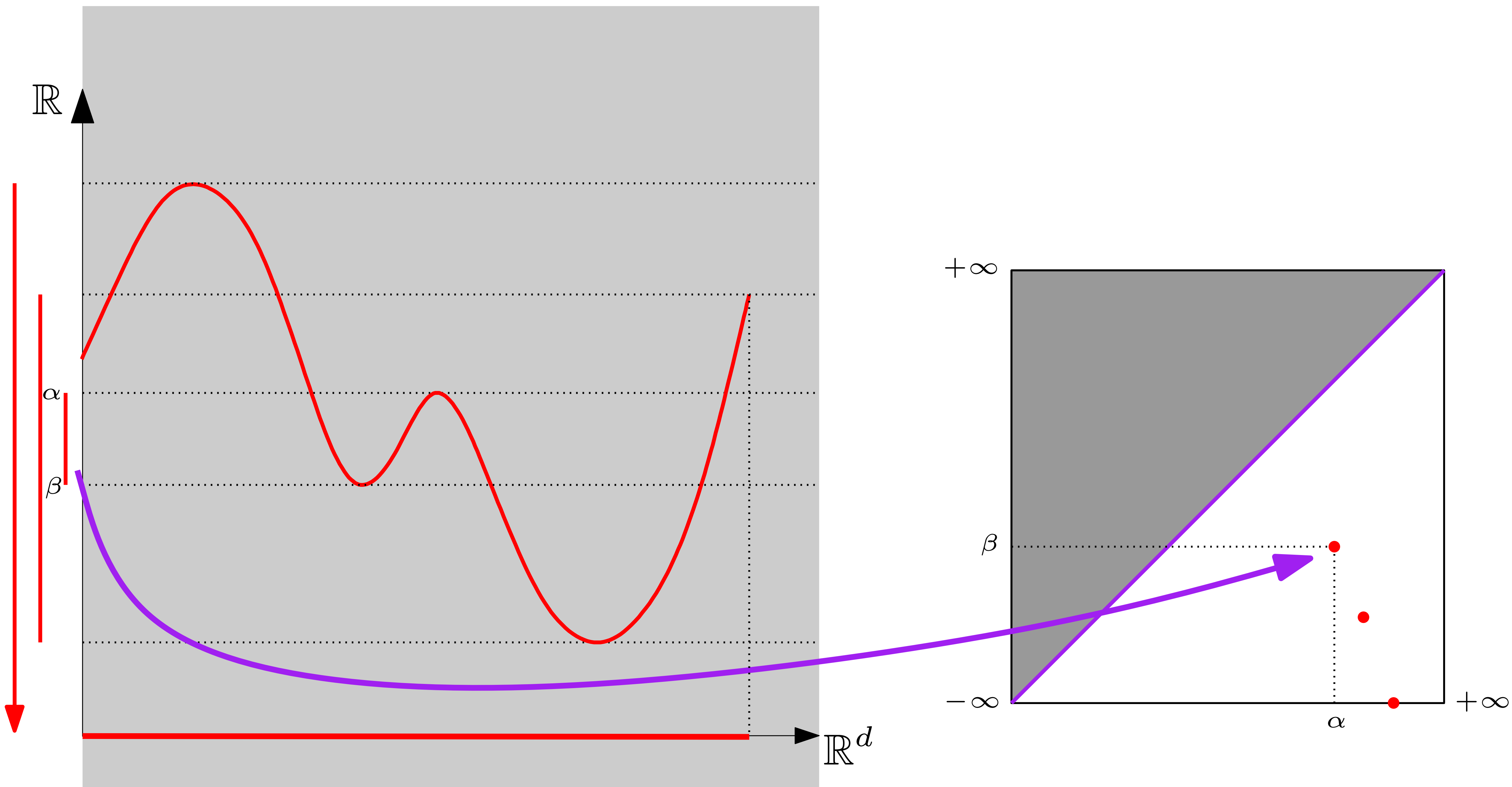
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.
- Finite set of intervals (barcode) encodes births/deaths of topological features.



Persistence for Mode Seeking

Given a probability density f:

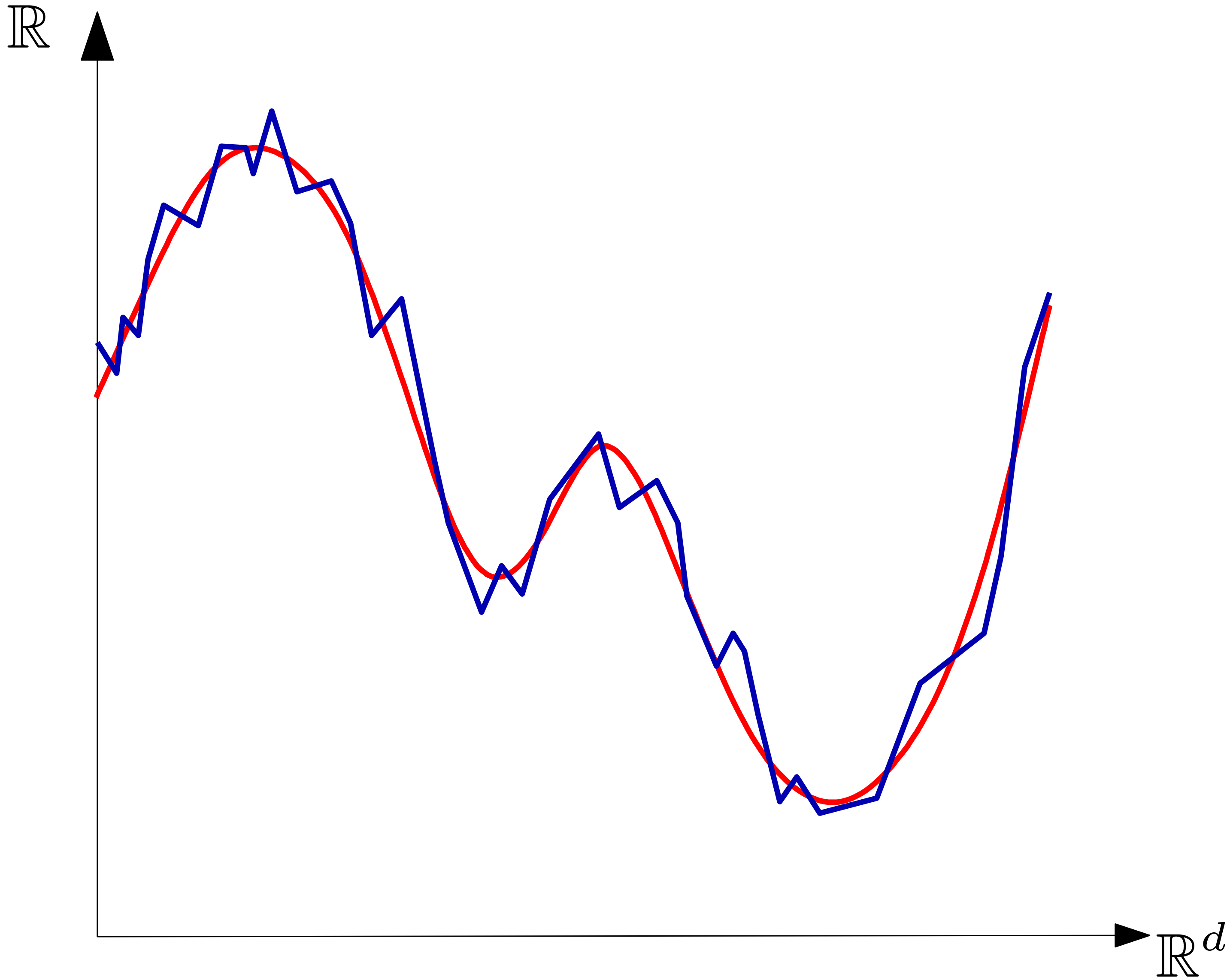
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$.
- Track evolution of topology throughout the family.
- Finite set of intervals (barcode) encodes births/deaths of topological features.

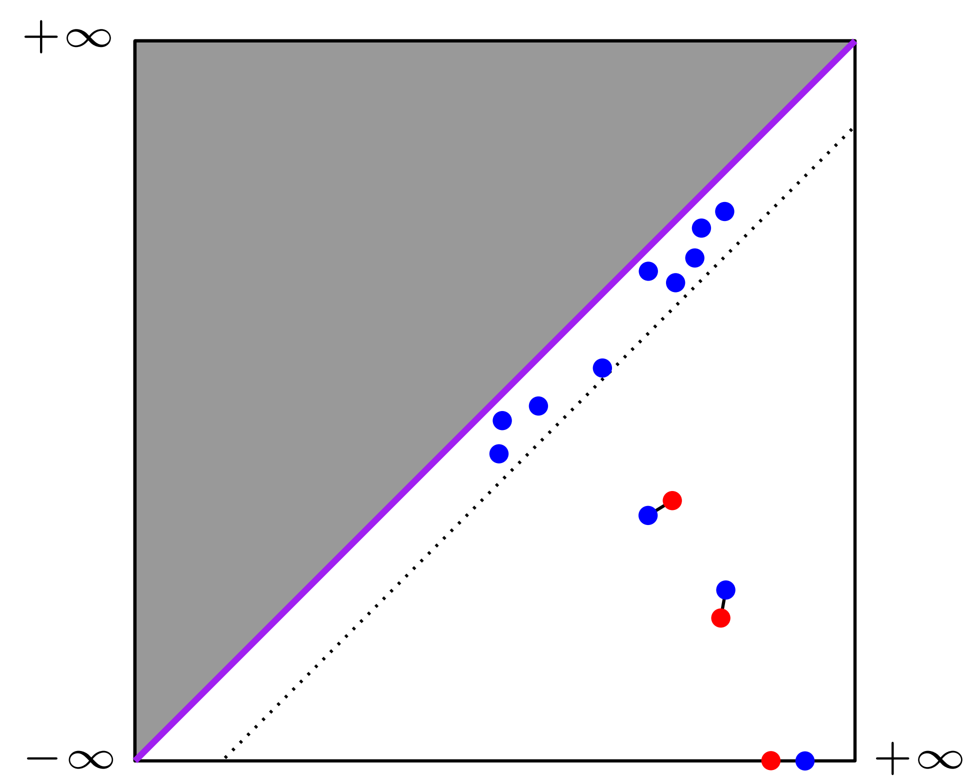


Persistence for Mode Seeking

Given an estimator $\hat{f}$:

Stability Theorem $\Rightarrow \mathrm{d}_B^\infty(\mathrm{Dg}\, f, \mathrm{Dg}\, \hat{f}) \leq \|f - \hat{f}\|_\infty$.

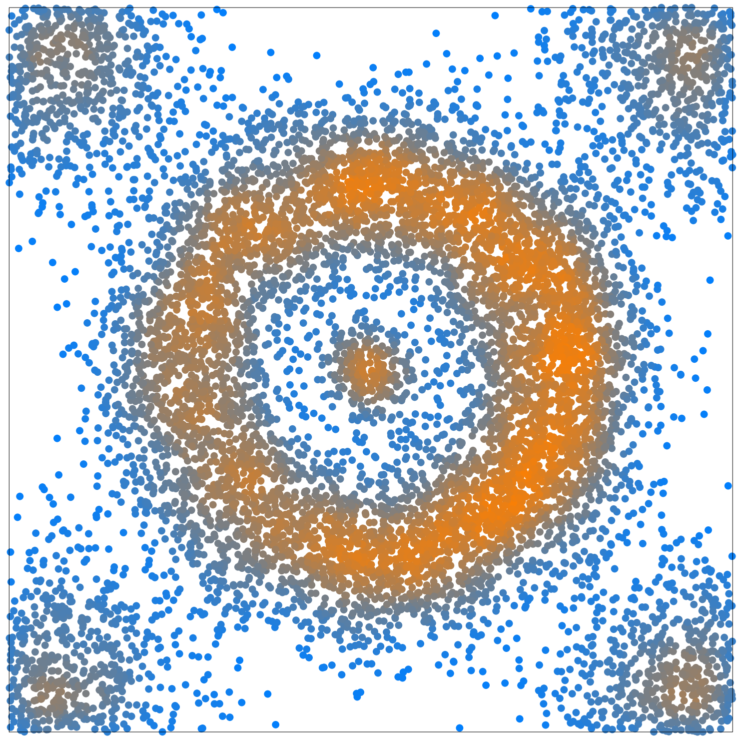




More precisely...

- Density estimator $\hat{f}$ defines an order on the point cloud

  (sort data points by **decreasing** estimated density values)



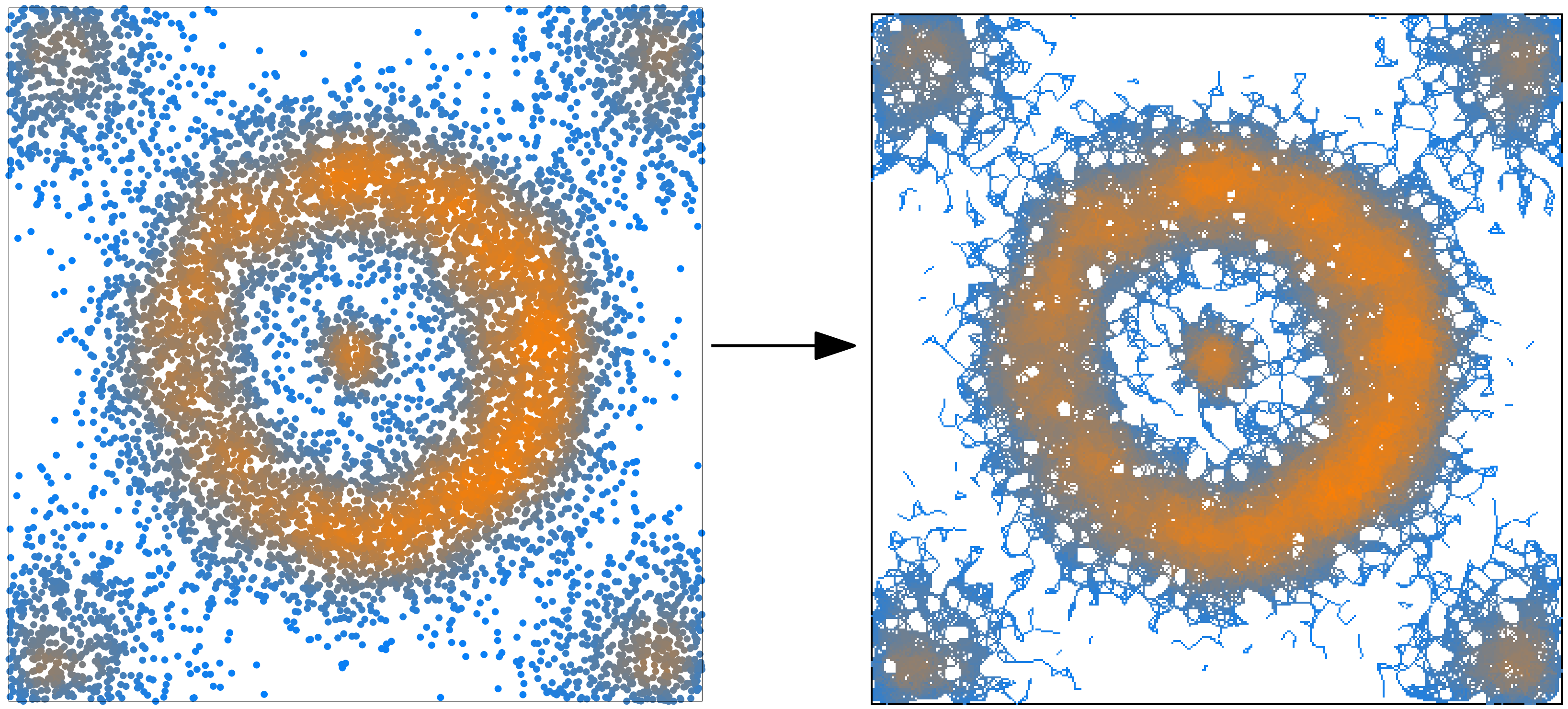
More precisely...

- Density estimator $\hat{f}$ defines an order on the point cloud

  (sort data points by **decreasing** estimated density values)

- Extend order to the graph edges $\rightarrow$ *upper-star filtration*

  $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$



More precisely...

- Density estimator $\hat{f}$ defines an order on the point cloud

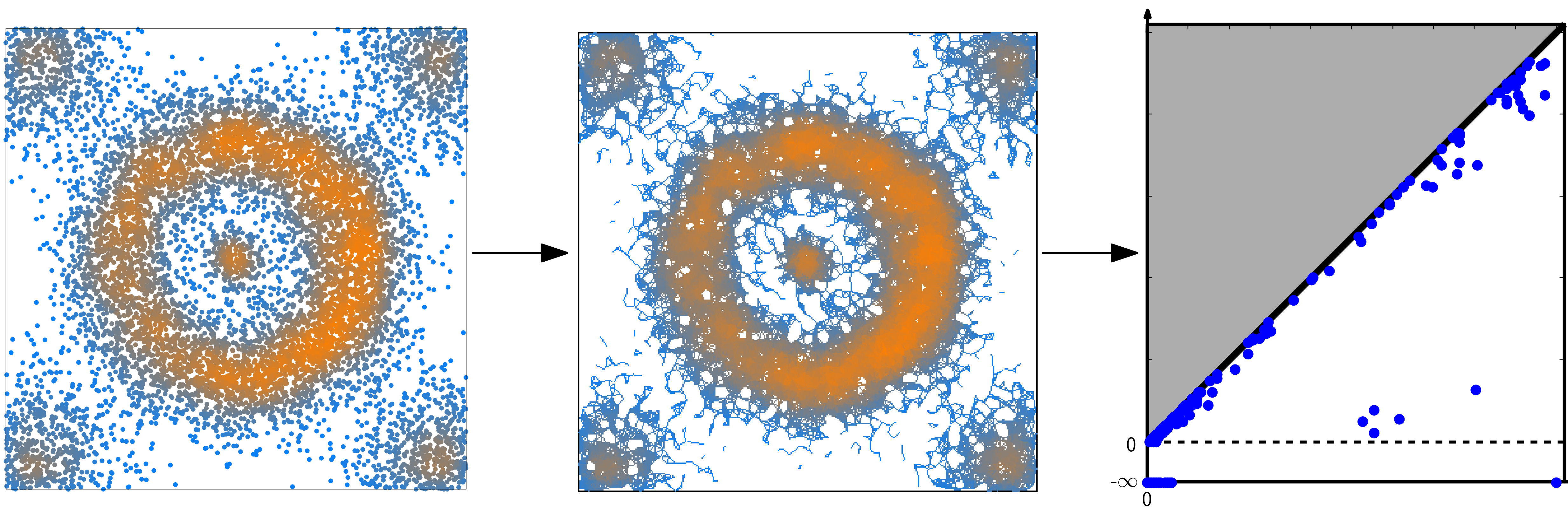
  (sort data points by **decreasing** estimated density values)

- Extend order to the graph edges $\rightarrow$ *upper-star filtration*

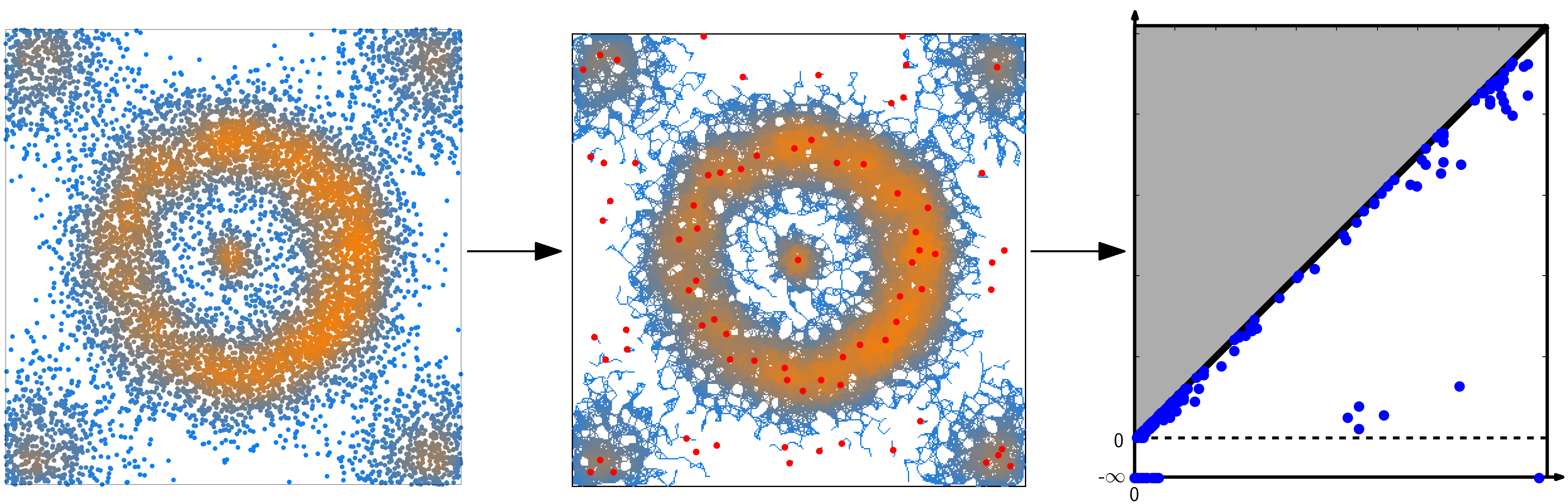
  $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

- Compute the 0-dimensional persistence diagram of this filtration

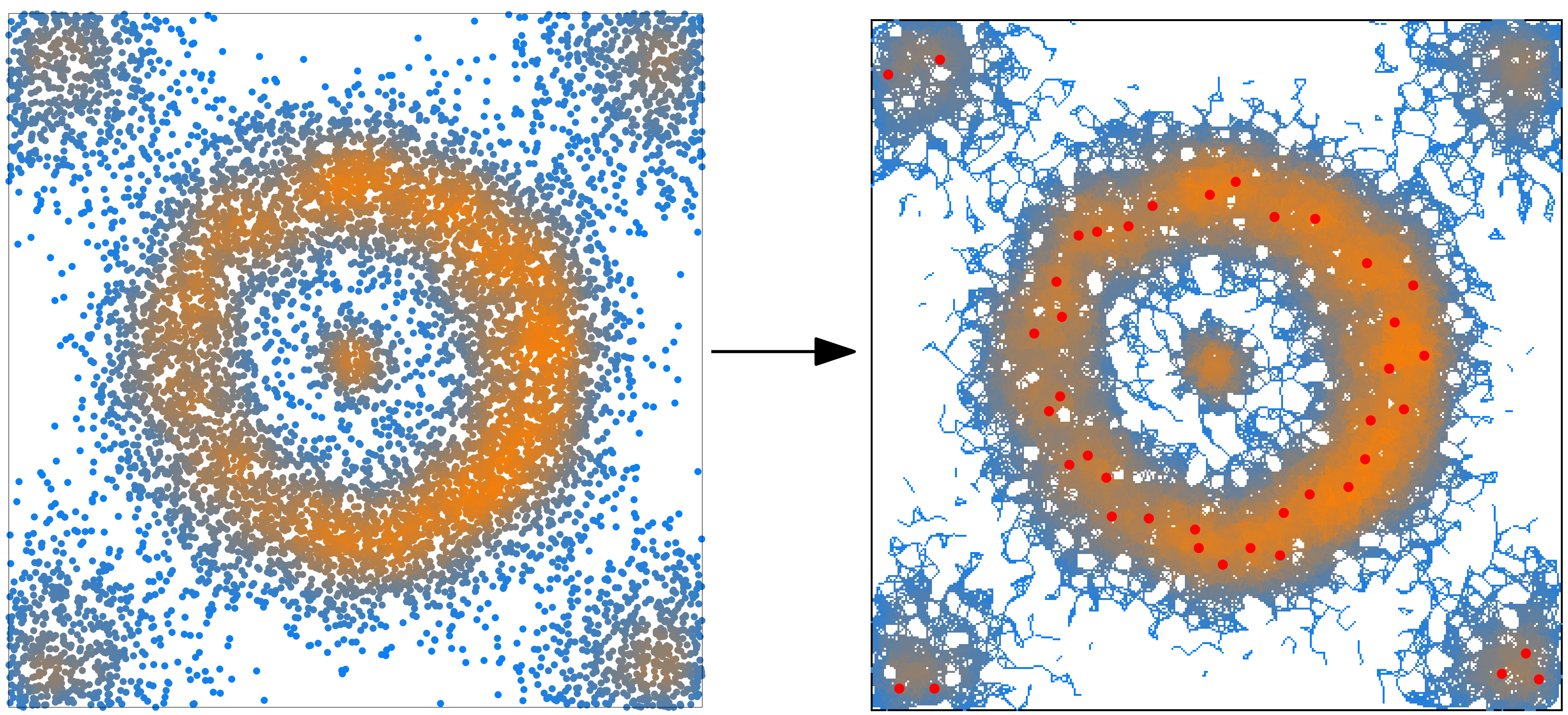
  (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)

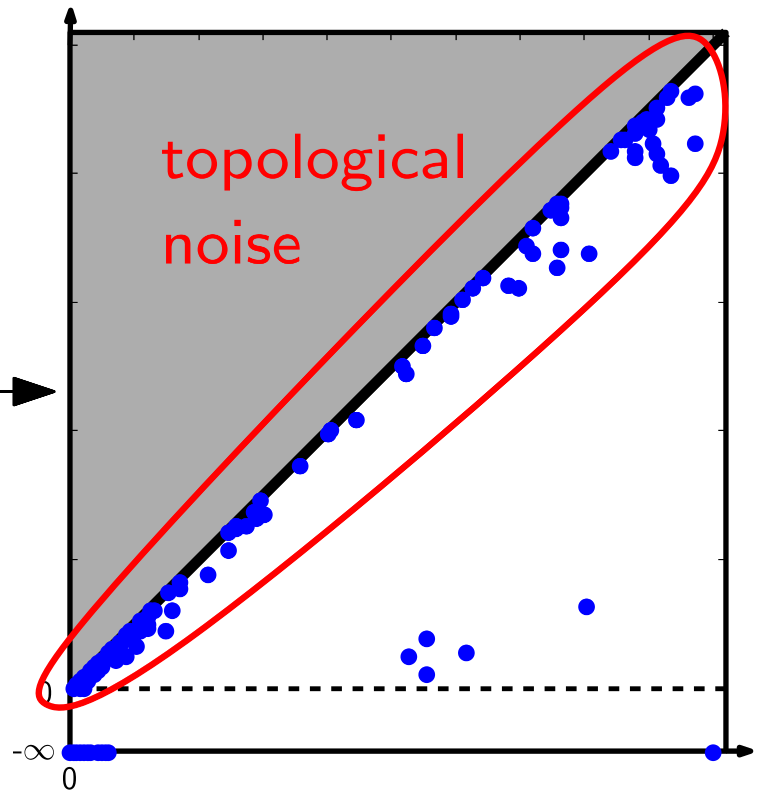


Estimating the Correct Number of Clusters

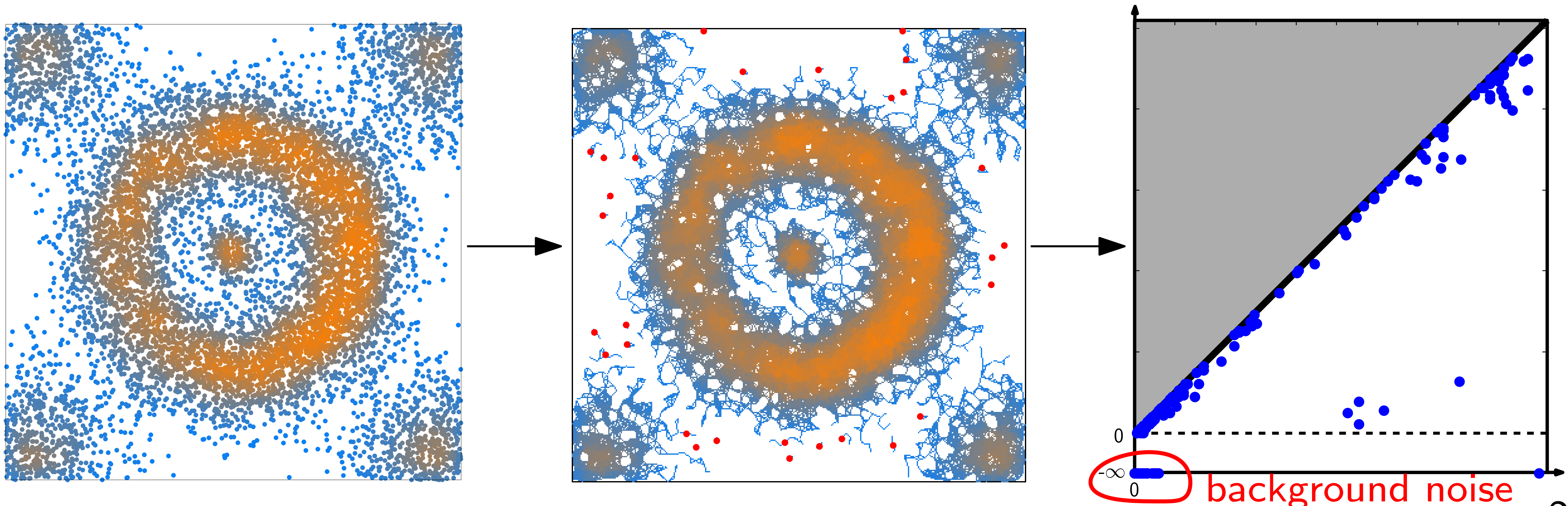


Estimating the Correct Number of Clusters

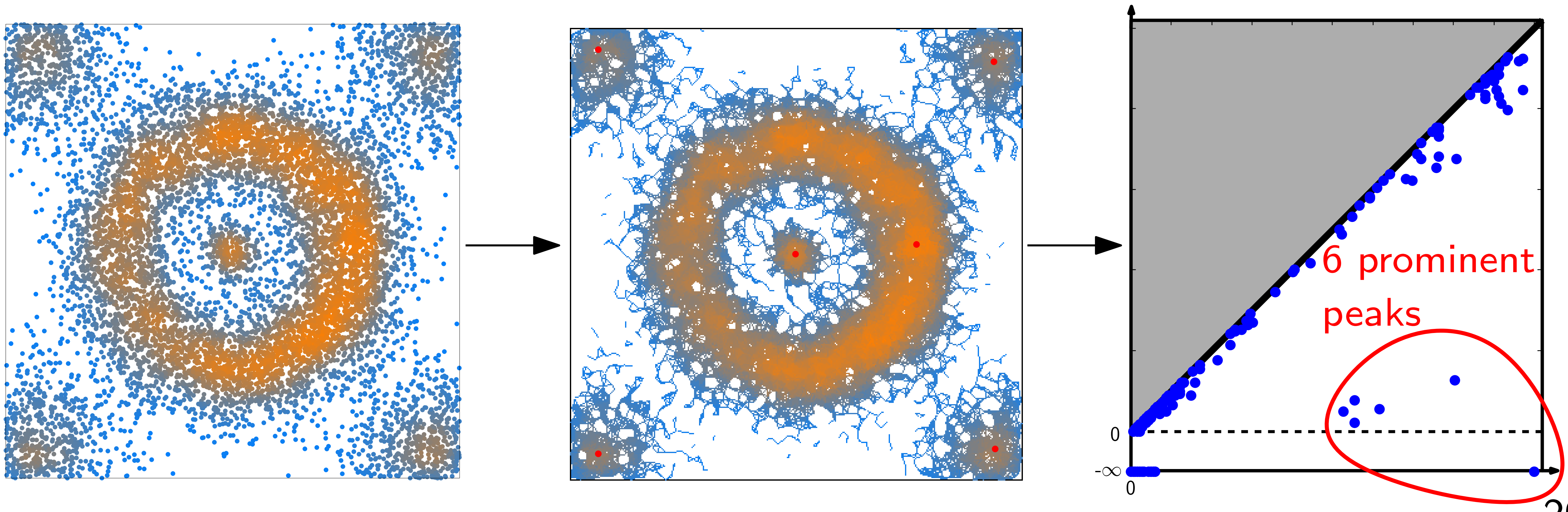




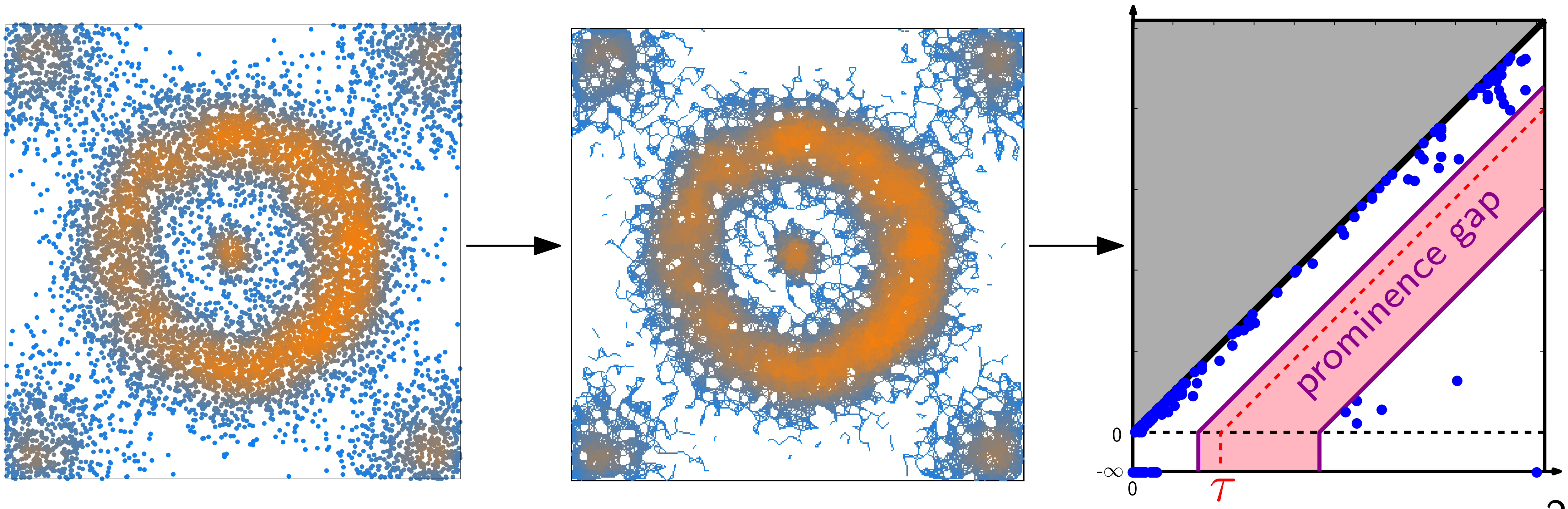
Estimating the Correct Number of Clusters



Estimating the Correct Number of Clusters



Estimating the Correct Number of Clusters



Estimating the Correct Number of Clusters

Hypotheses:

- $f : \mathbb{R}^d \to \mathbb{R}$ a c-Lipschitz probability density function,
- $P \subset \mathbb{R}^d$ a finite set of n points sampled i.i.d. according to f,
- $\hat{f} : P \to \mathbb{R}$ a density estimator such that $\eta := \max_{p \in P} |\hat{f}(p) - f(p)| < \Pi/5$,
- $G = (P, E)$ the δ-neighborhood graph for some positive $\delta < \frac{\Pi - 5\eta}{5c}$.

Note: Π is the prominence of the least prominent peak of f

Estimating the Correct Number of Clusters

Hypotheses:

- $f : \mathbb{R}^d \to \mathbb{R}$ a c-Lipschitz probability density function,
- $P \subset \mathbb{R}^d$ a finite set of n points sampled i.i.d. according to f,
- $\hat{f} : P \to \mathbb{R}$ a density estimator such that $\eta := \max_{p \in P} |\hat{f}(p) - f(p)| < \Pi/5$,
- $G = (P, E)$ the δ-neighborhood graph for some positive $\delta < \frac{\Pi - 5\eta}{5c}$.

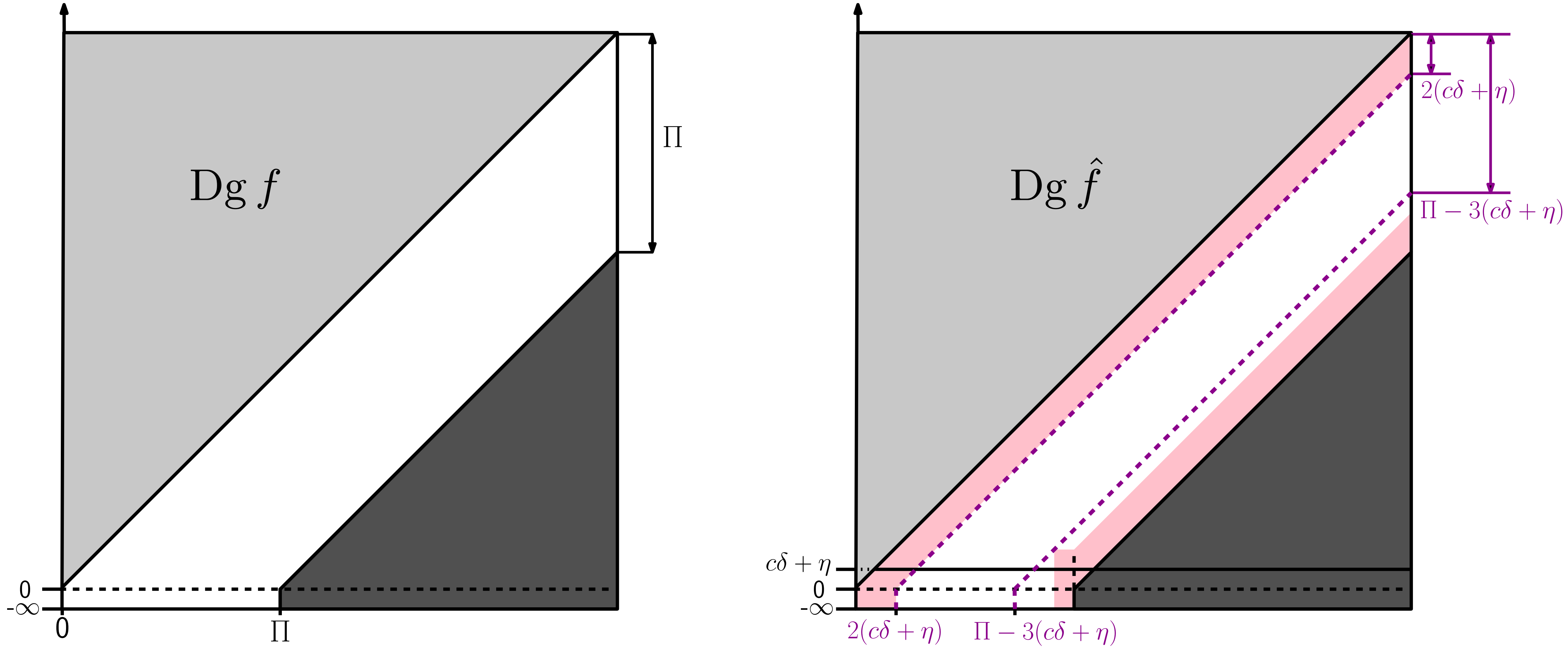
Note: Π is the prominence of the least prominent peak of f

Conclusion:

For any choice of τ such that $2(c\delta + \eta) < \tau < \Pi - 3(c\delta + \eta)$, the number of clusters computed by the algorithm is equal to the number of peaks of f with probability at least $1 - e^{-\Omega(n)}$.

(the Ω notation hides factors depending on c, δ)

Estimating the Correct Number of Clusters

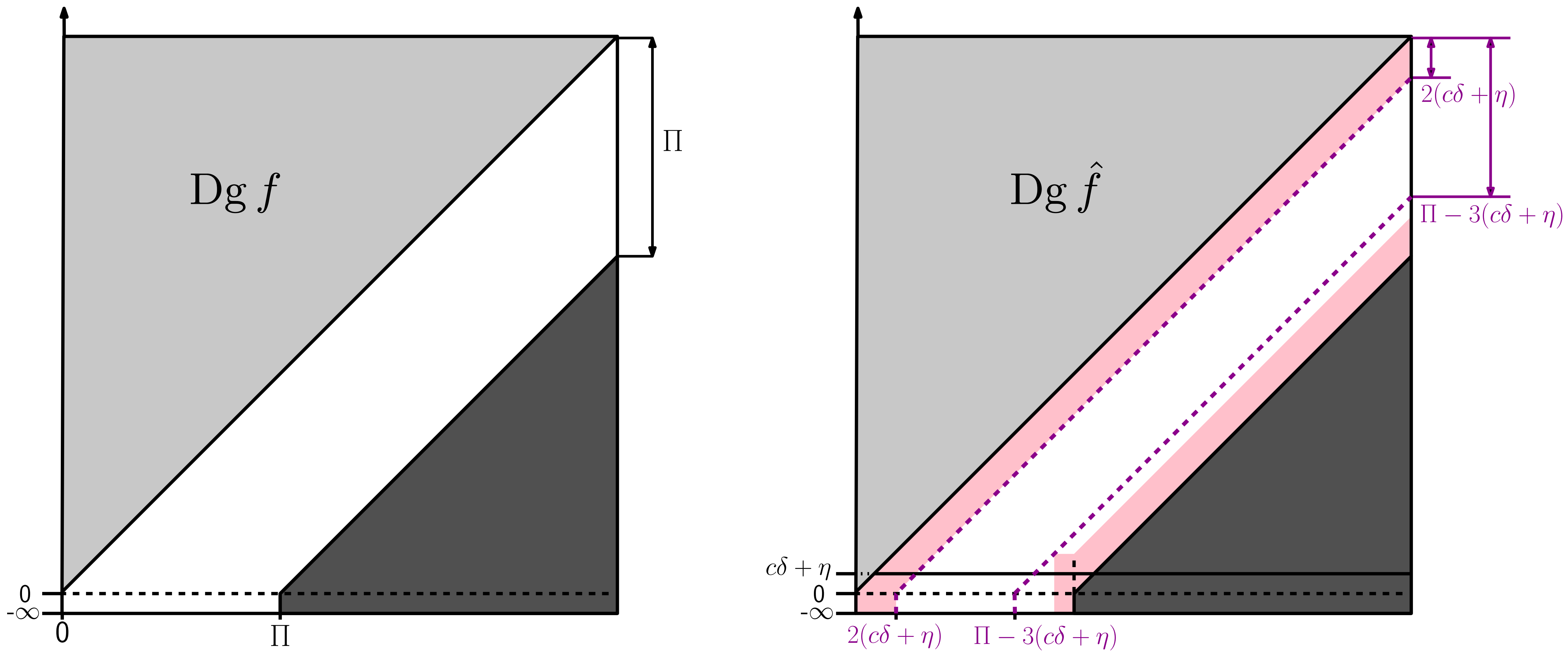


Conclusion:

For any choice of τ such that $2(c\delta+\eta) < \tau < \Pi - 3(c\delta+\eta)$, the number of clusters computed by the algorithm is equal to the number of peaks of f with probability at least $1 - e^{-\Omega(n)}$.

(the Ω notation hides factors depending on c, δ)

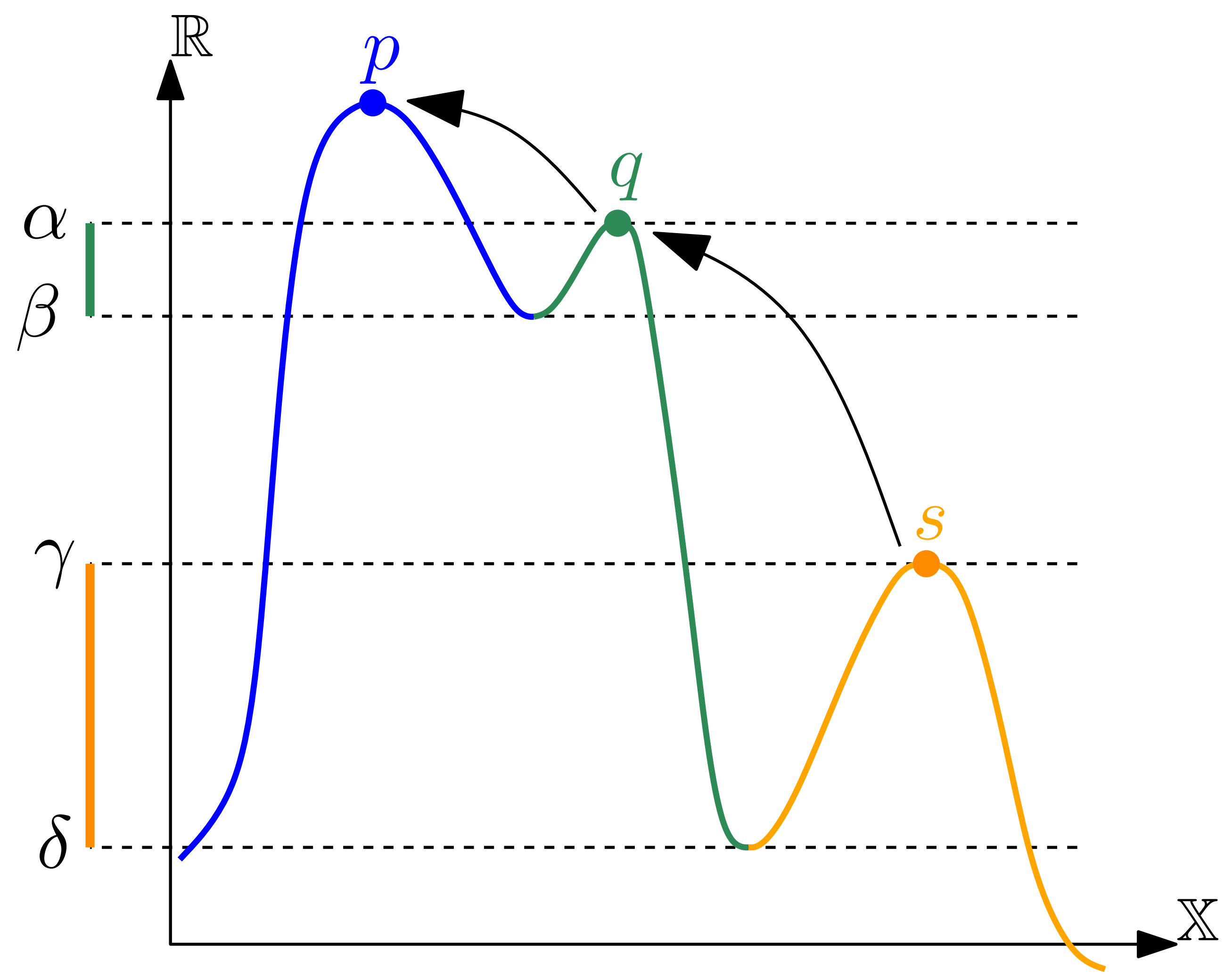
Estimating the Correct Number of Clusters



Proof's main ingredient: stability theorem for persistence diagrams

Merging Clusters

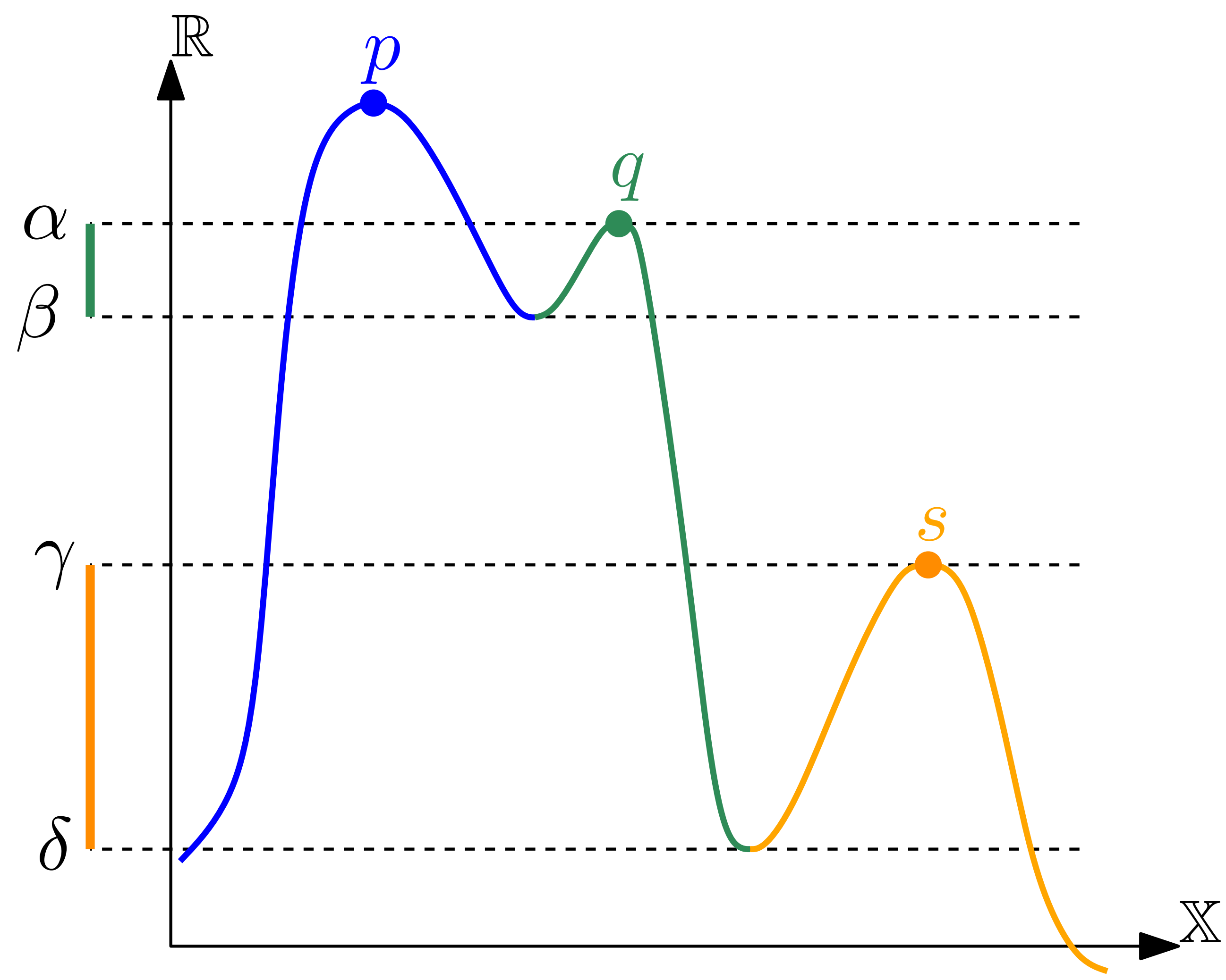
- degree-0 persistence algo. builds a hierarchy of the peaks of $\hat{f}$ (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)



Merging Clusters

- degree-0 persistence algo. builds a hierarchy of the peaks of $\hat{f}$ (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)
- given a fixed threshold $\tau \geq 0$, only merge those clusters of prominence $< \tau$

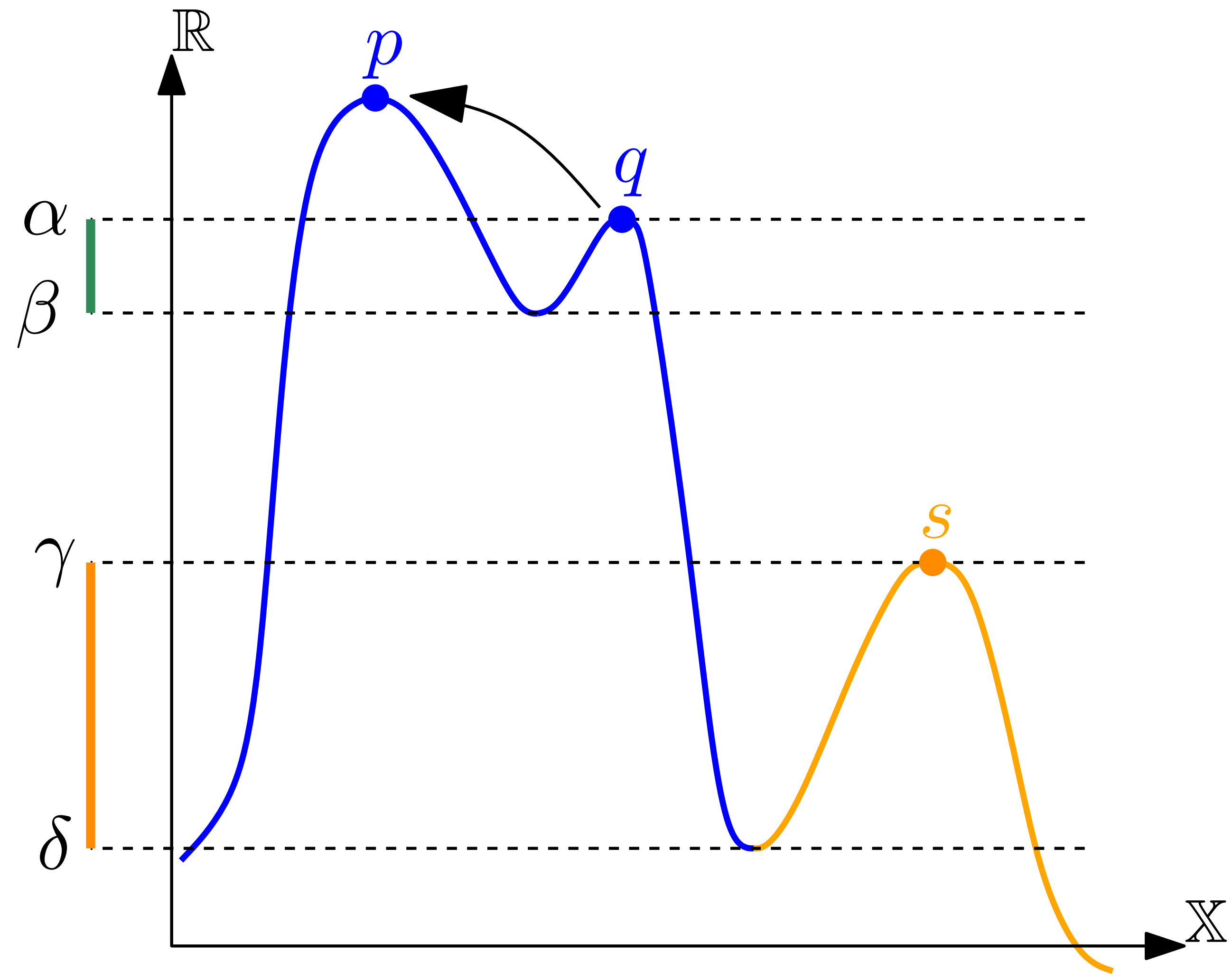
$$0 \leq \tau \leq \alpha - \beta$$



Merging Clusters

- degree-0 persistence algo. builds a hierarchy of the peaks of $\hat{f}$ (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)
- given a fixed threshold $\tau \geq 0$, only merge those clusters of prominence $< \tau$

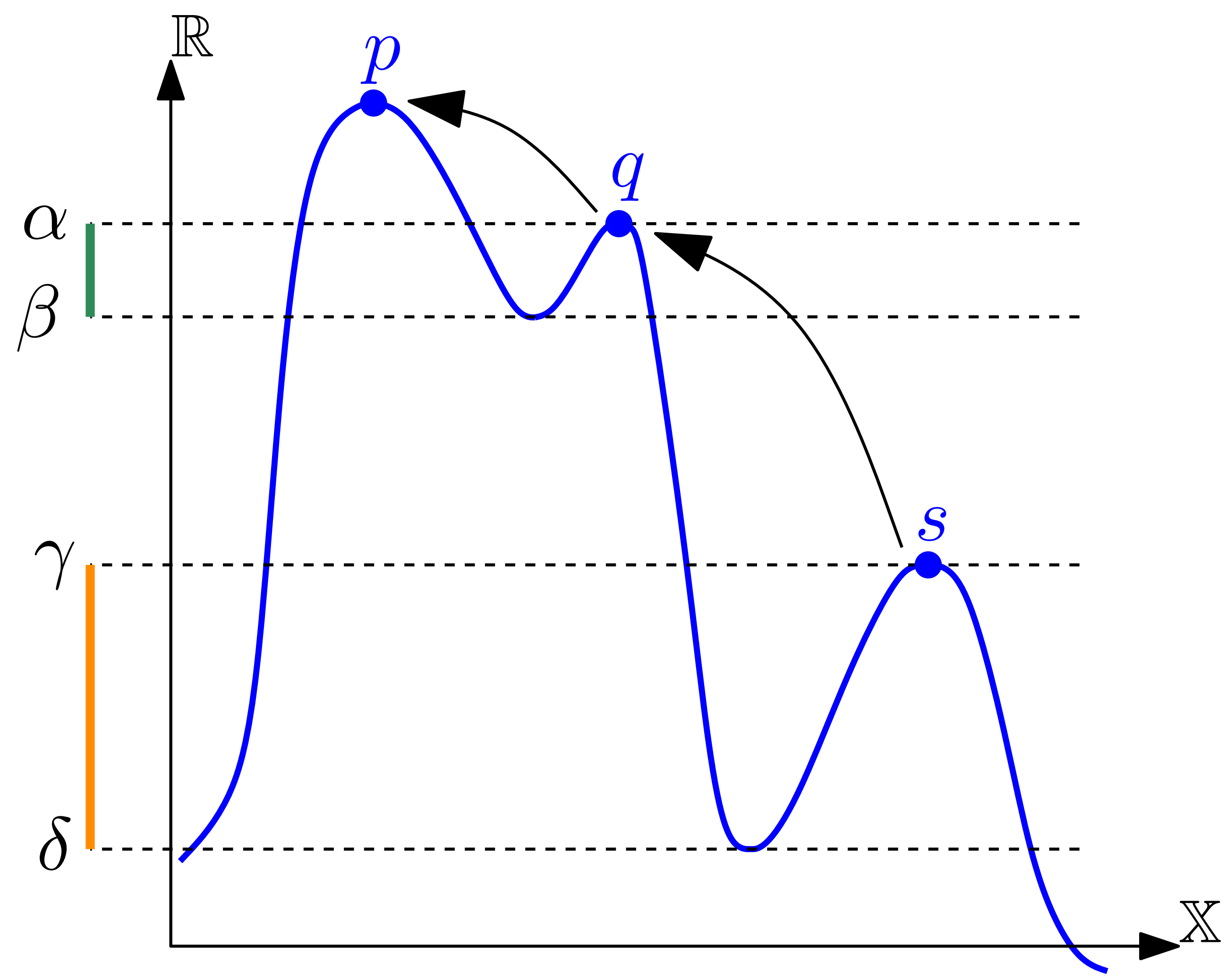
$$\alpha - \beta < \tau \leq \gamma - \delta$$



Merging Clusters

- degree-0 persistence algo. builds a hierarchy of the peaks of $\hat{f}$ (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)
- given a fixed threshold $\tau \geq 0$, only merge those clusters of prominence $< \tau$

$$\gamma - \delta < \tau \leq +\infty$$



Pseudo-code:

Input: simple graph G with n vertices, n-dimensional vector $\hat{f}$, real parameter $\tau \geq 0$.

Sort the vertex indices $\{1, 2, \cdots, n\}$ so that $\hat{f}(1) \geq \hat{f}(2) \geq \cdots \geq \hat{f}(n)$;
Initialize a union-find data structure $\mathcal{U}$ and two vectors g, r of size n;

for $i = 1$ to n **do**
  Let $\mathcal{N}$ be the set of neighbors of i in G that have indices lower than i;
  if $\mathcal{N} = \emptyset$ *// vertex i is a peak of $\hat{f}$ within G*
    Create a new entry e in $\mathcal{U}$ and attach vertex i to it;
    $r(e) \leftarrow i$ *// $r(e)$ stores the root vertex associated with the entry e*
  else *// vertex i is not a peak of $\hat{f}$ within G*
    $g(i) \leftarrow \mathrm{argmax}_{j \in \mathcal{N}} \hat{f}(j)$ *// $g(i)$ stores the approximate gradient at vertex i*
    $e_i \leftarrow \mathcal{U}.\mathtt{find}(g(i))$;
    Attach vertex i to the entry e_i;

graph-based hill-climbing (1976)

  for $j \in \mathcal{N}$ **do**
    $e \leftarrow \mathcal{U}.\mathtt{find}(j)$;
    if $e \neq e_i$ and $\min\{\hat{f}(r(e)),\ \hat{f}(r(e_i))\} < \hat{f}(i) + \tau$
      $\mathcal{U}.\mathtt{union}(e,\ e_i)$;
      $r(e \cup e_i) \leftarrow \mathrm{argmax}_{\{r(e),\ r(e_i)\}} \hat{f}$;
      $e_i \leftarrow e \cup e_i$;

cluster merges with persistence (2013)

Output: the collection of entries e of $\mathcal{U}$ such that $\hat{f}(r(e)) \geq \tau$.

Complexity of the Algorithm

Given a neighborhood graph with n vertices (with density values) and m edges:

1. the algorithm sorts the vertices by decreasing density values,

2. the algorithm makes a single pass through the vertex set, creating the spanning forest and merging clusters on the fly using a union-find data structure.

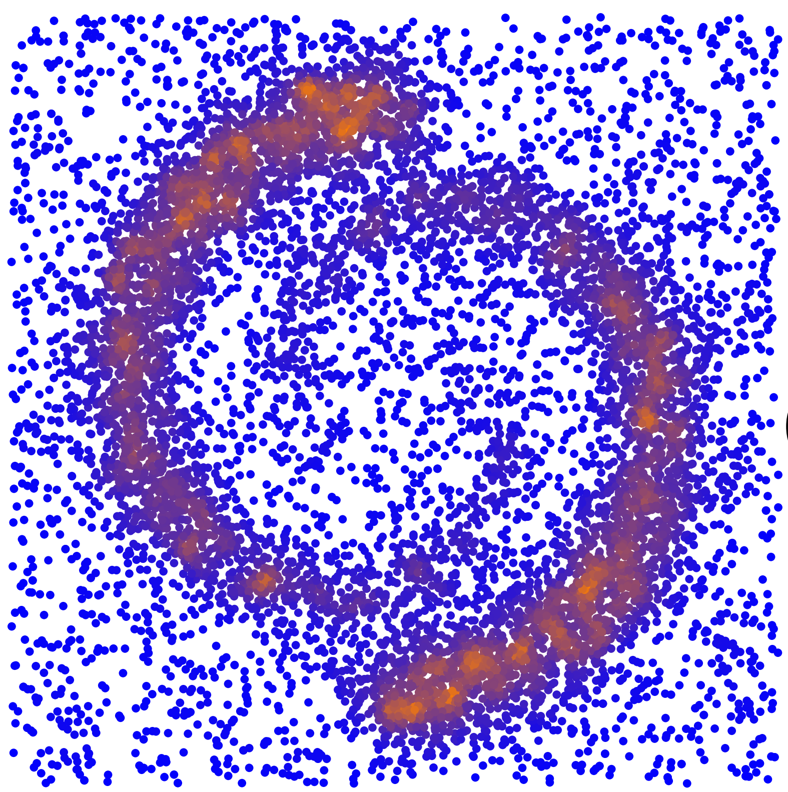
$\rightarrow$ Running time: $O(n \log n + (n+m)\alpha(n))$

$\rightarrow$ Space complexity: $O(n+m)$

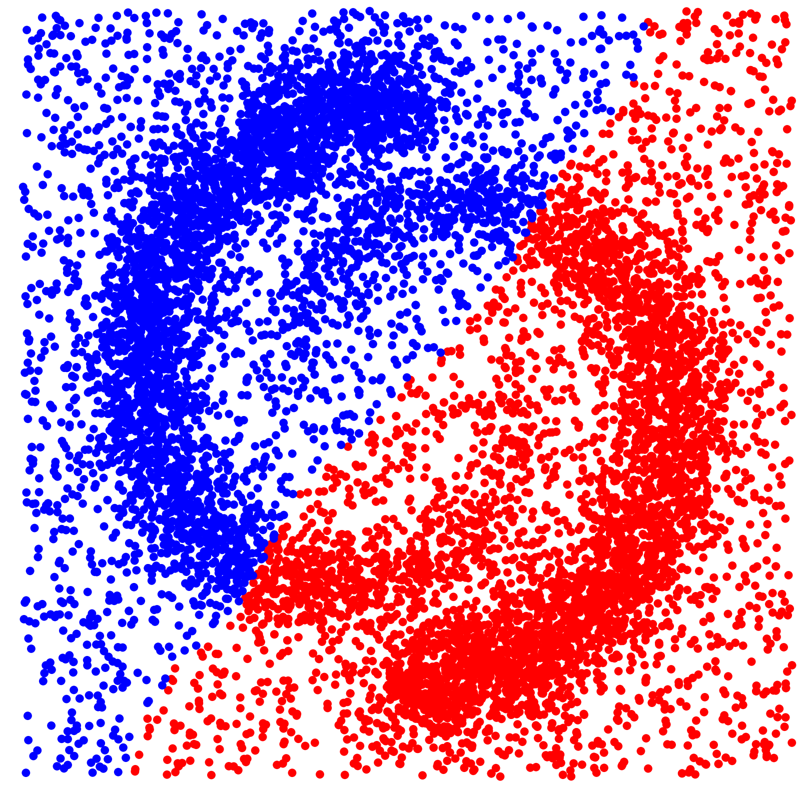
$\rightarrow$ Main memory usage: $O(n)$

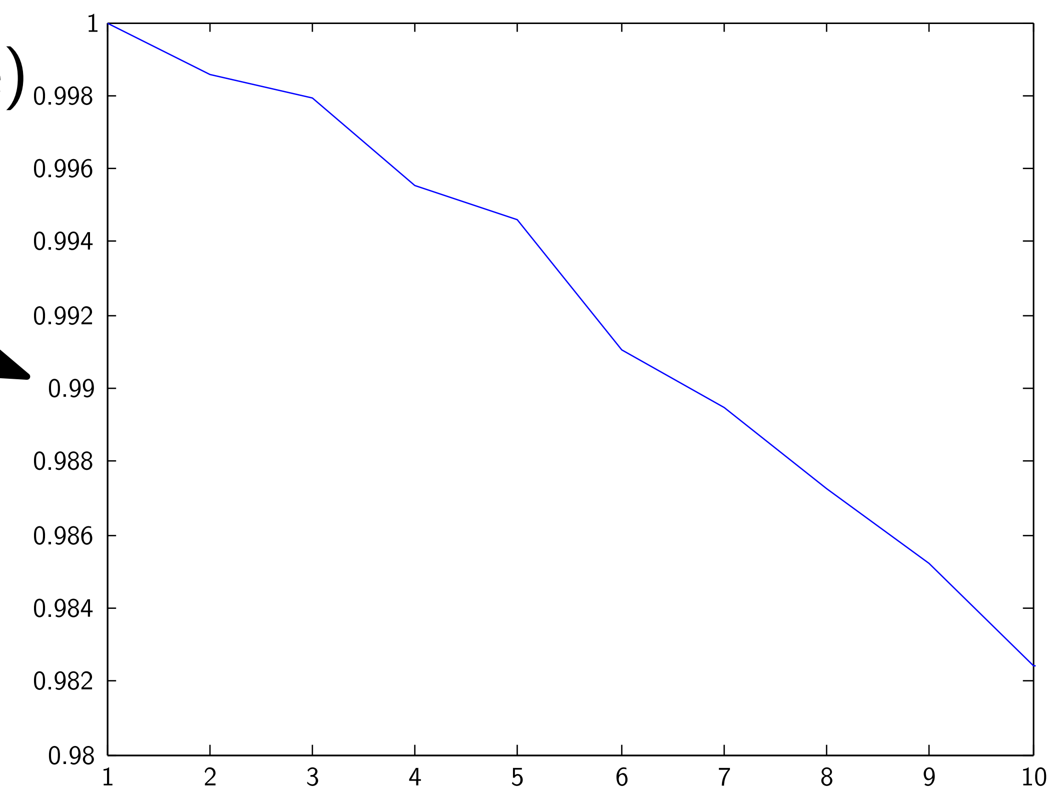
Experimental Results

Synthetic Data



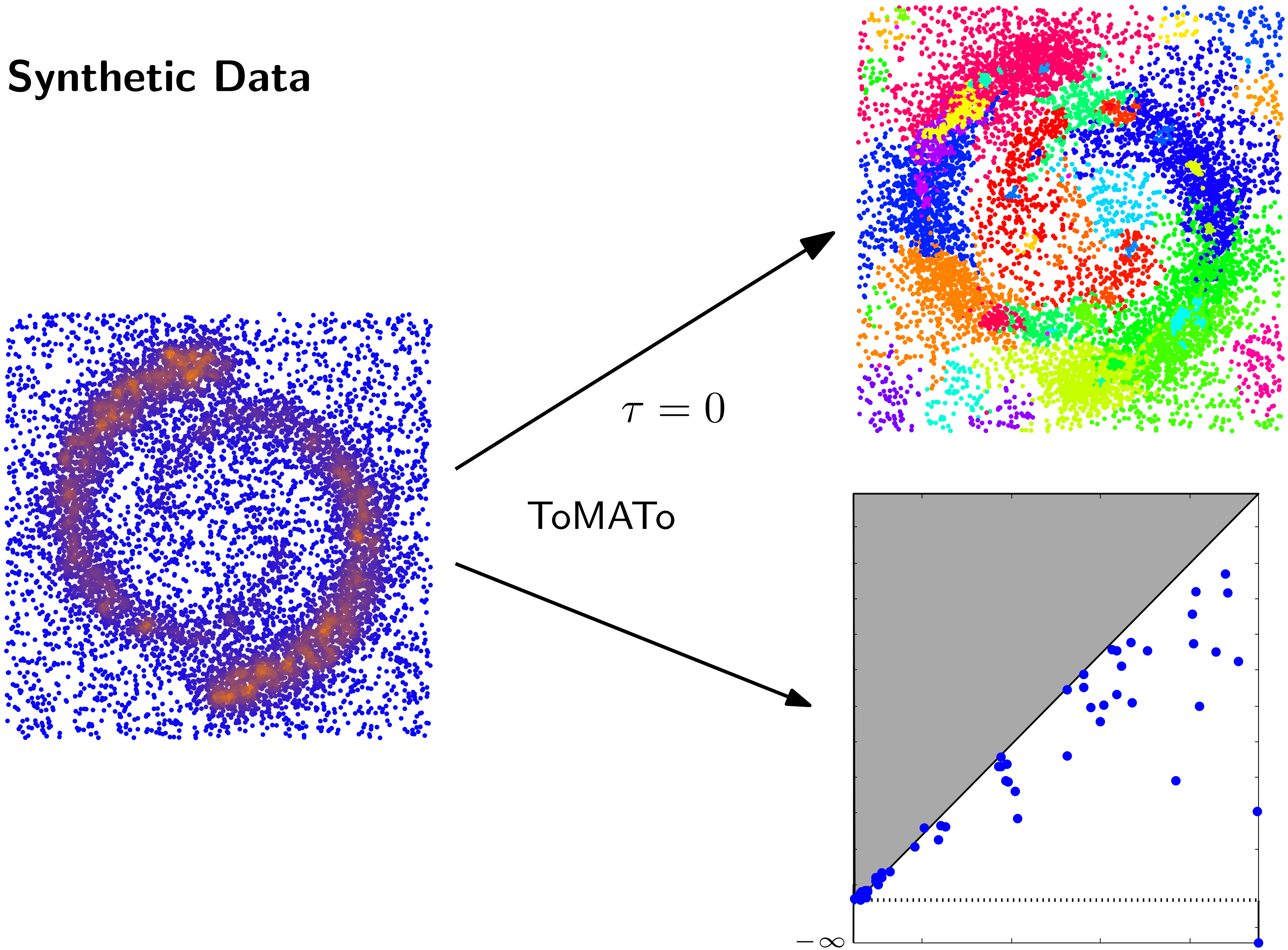
Spectral clustering
(k-means in eigenspace)





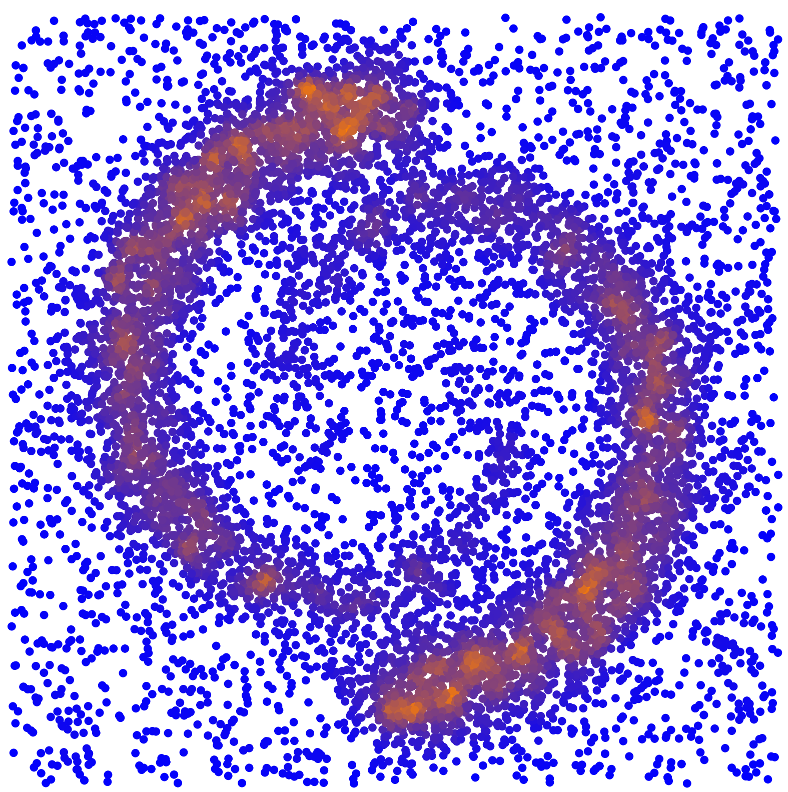
Experimental Results

Synthetic Data

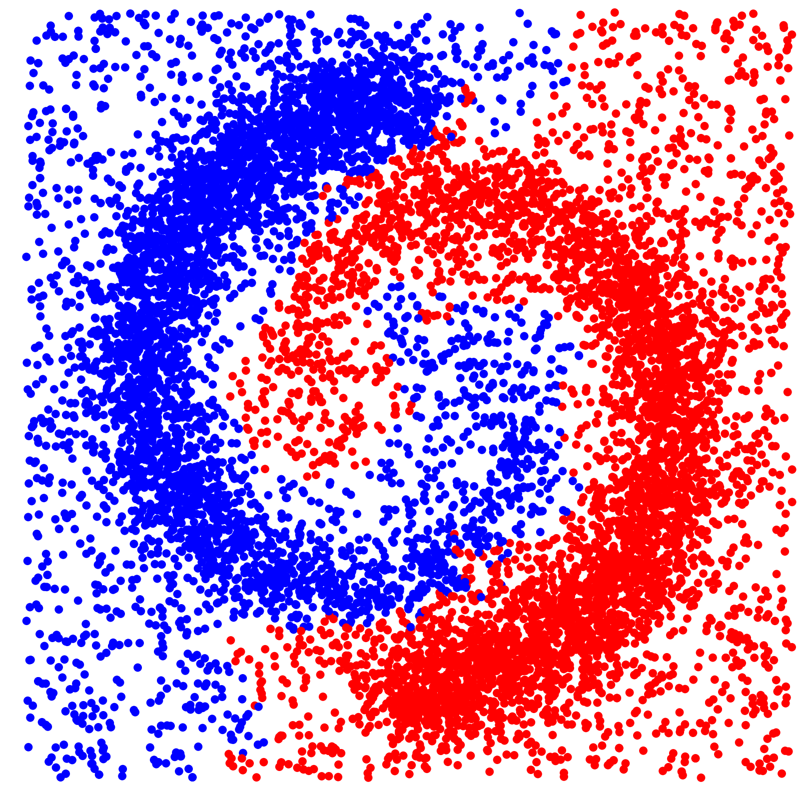


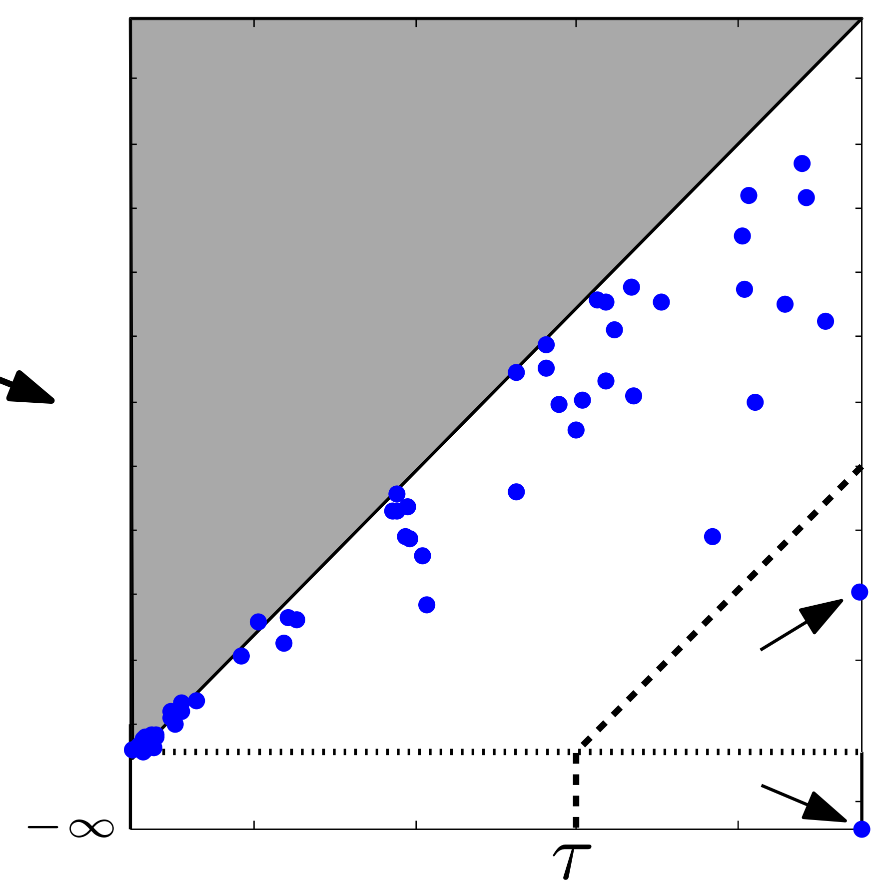
Experimental Results

Synthetic Data



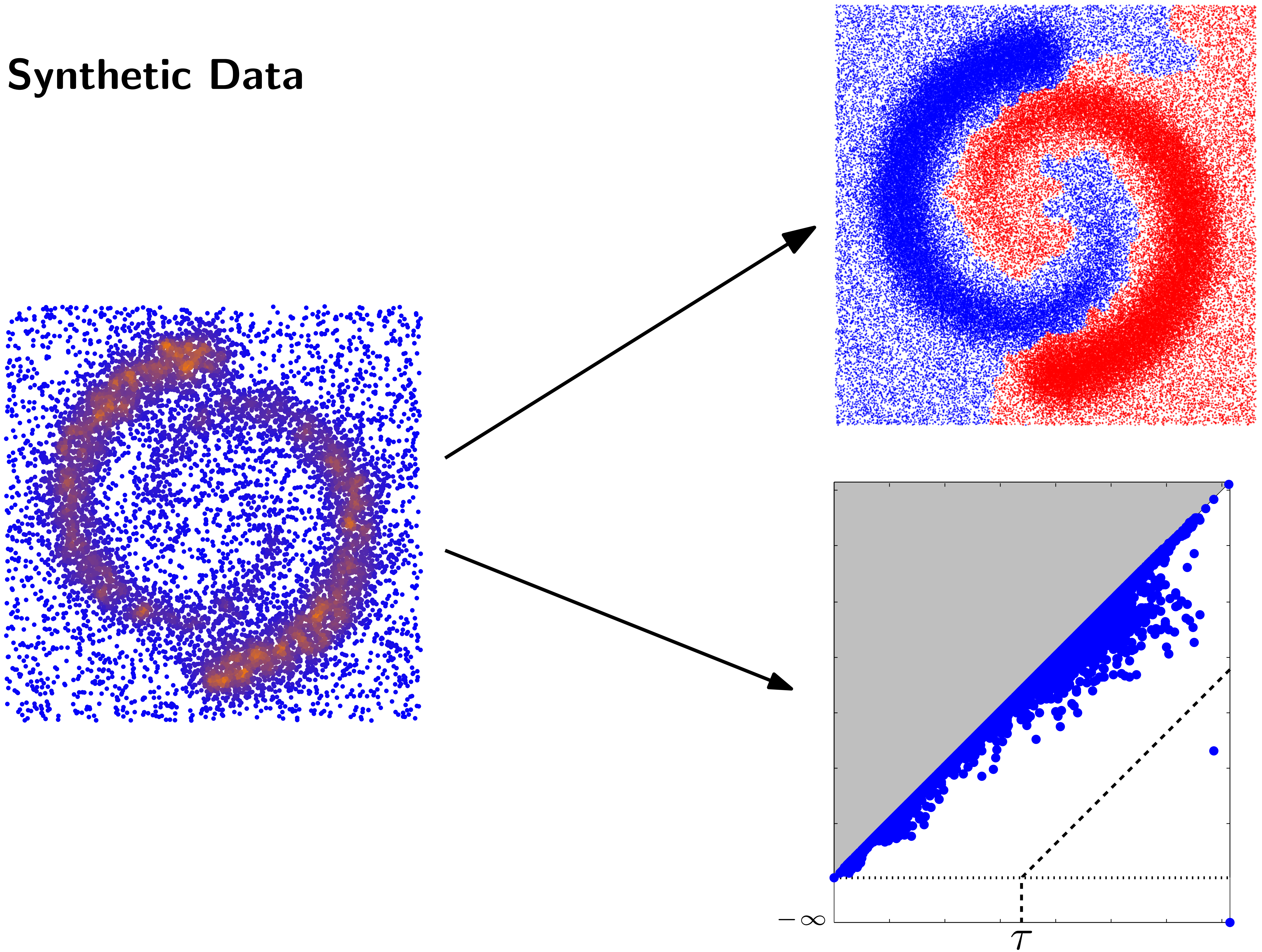
ToMATo





Experimental Results

Synthetic Data

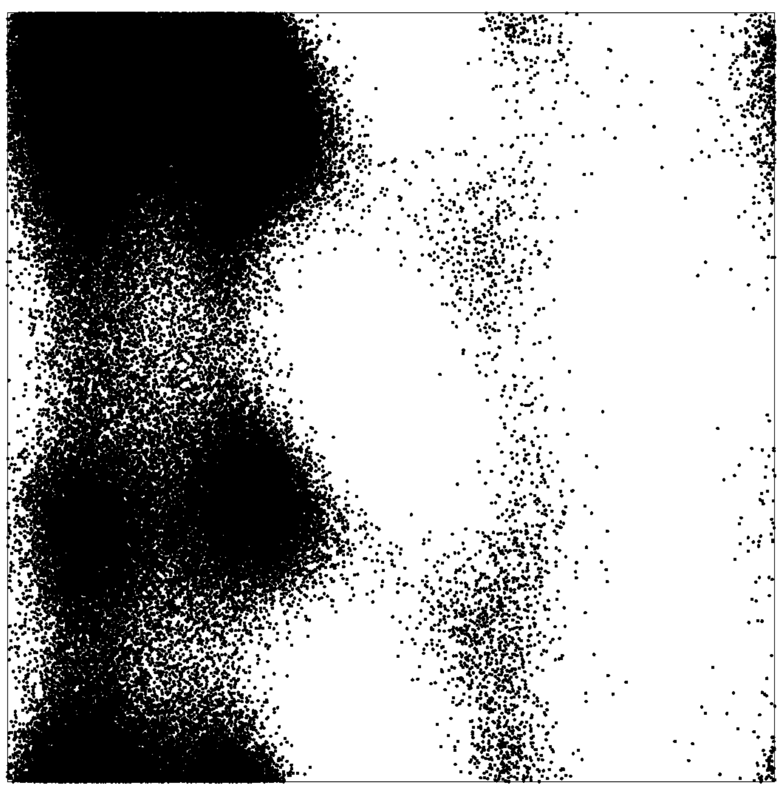


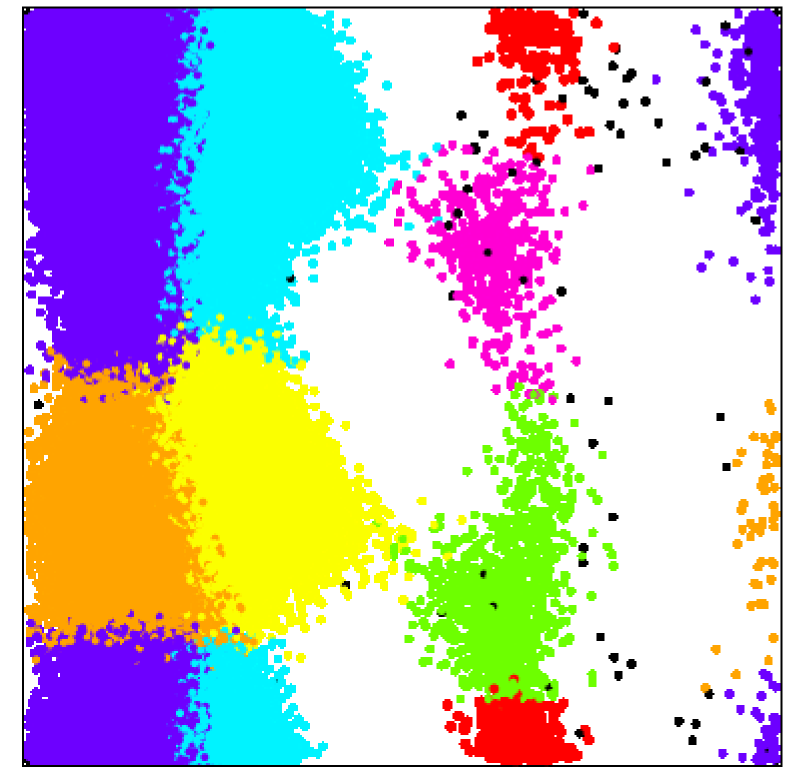
Experimental Results

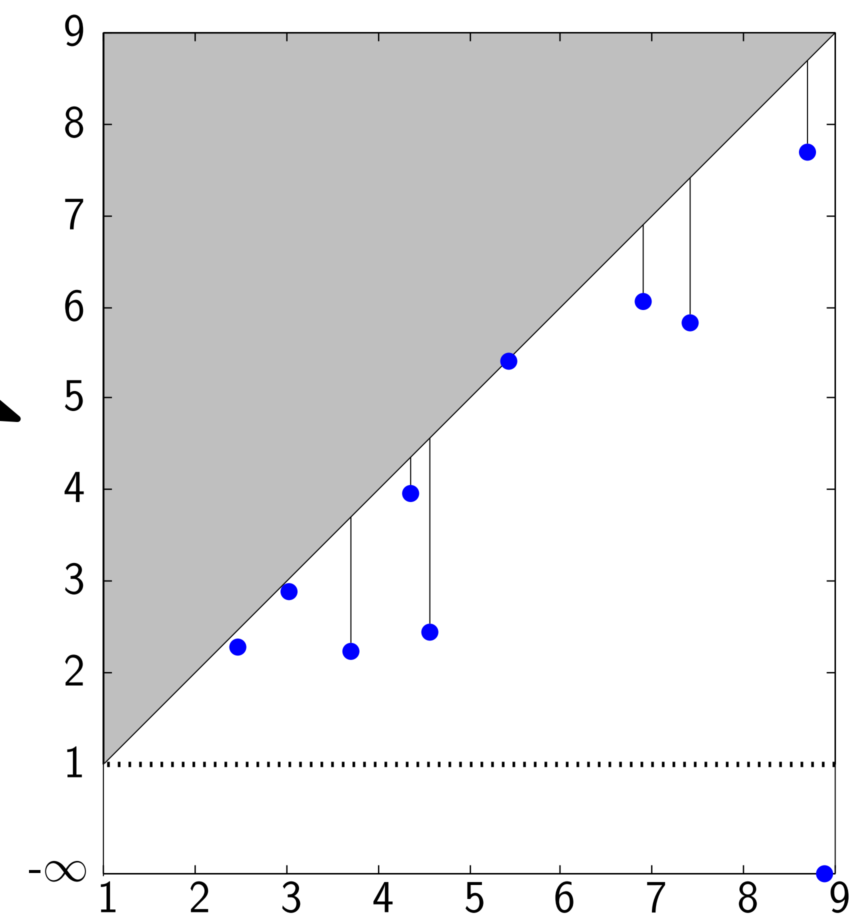
Biological Data

Alanine-Dipeptide conformations ($\mathbb{R}^{21}$)

RMSD distance (non-Euclidean)







Common belief: 6 metastable states

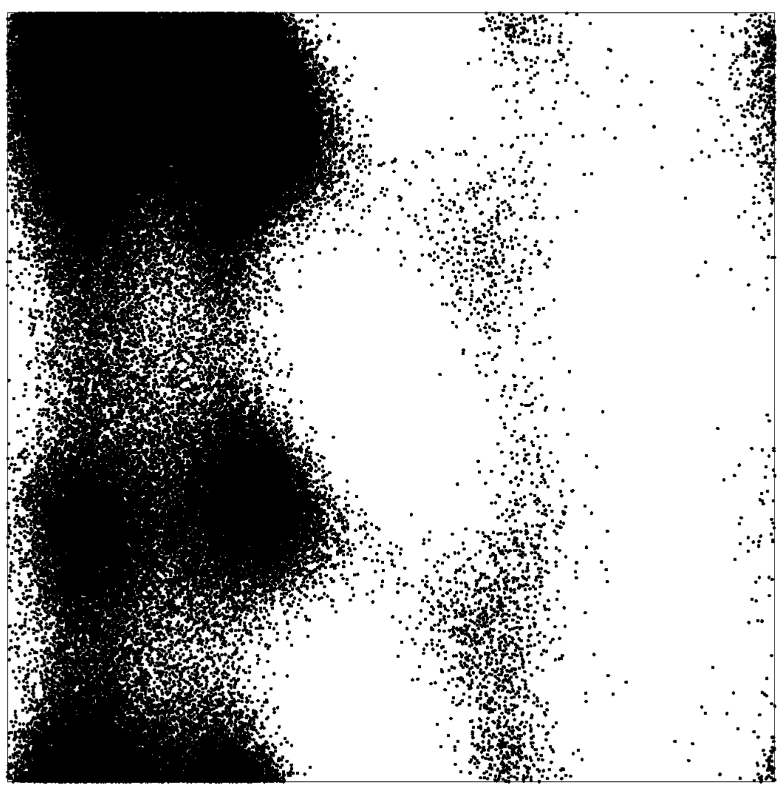
PD shows anywhere between 4 and 7 clusters

Experimental Results

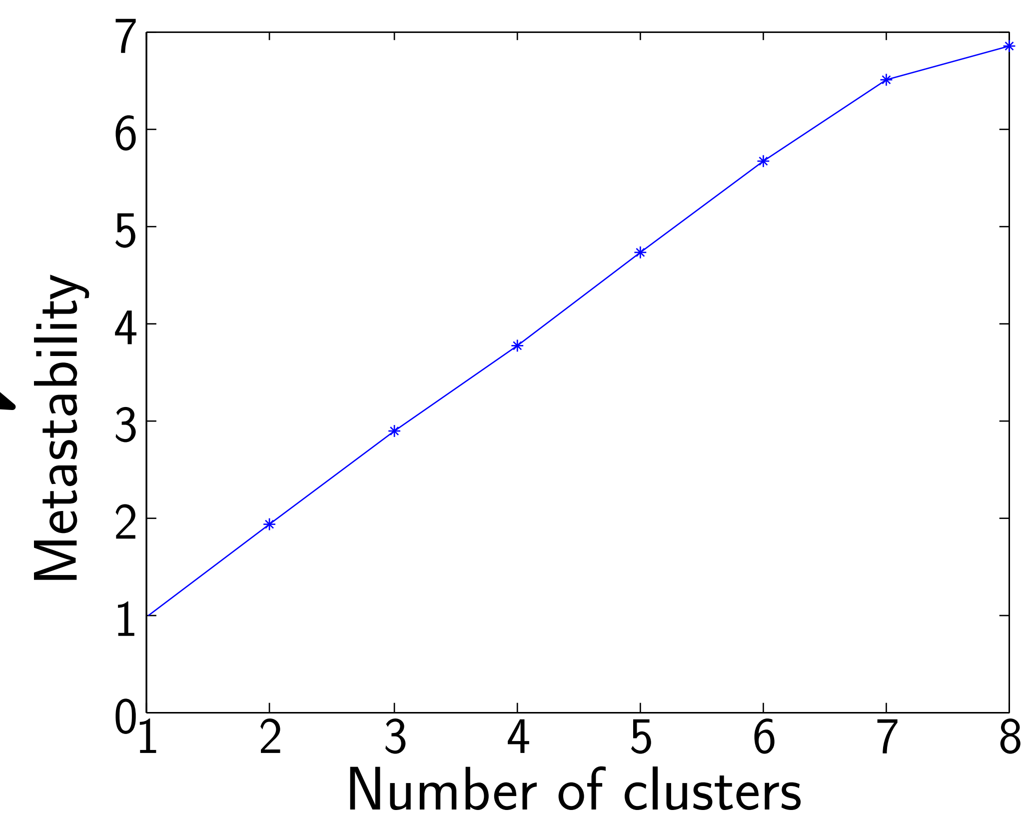
Biological Data

Alanine-Dipeptide conformations ($\mathbb{R}^{21}$)

RMSD distance (non-Euclidean)



Rank	Prominence	Metastability
1	$+\infty$	0.99982
2	3827	1.91865
3	1334	2.8813
4	557	3.76217
5	85	4.73838
6	32	5.65553
7	26	6.50757
8	7.2	6.8193
9	3.0	-
10	2.2	-



Common belief: 6 metastable states

PD shows anywhere between 4 and 7 clusters

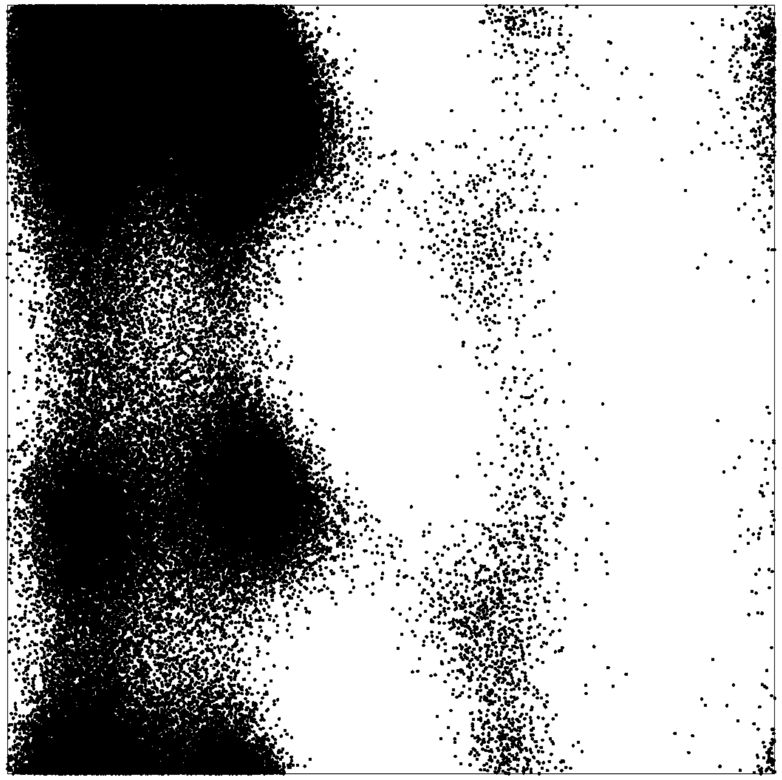
Measures of metastability confirm this insight

Experimental Results

Biological Data

Alanine-Dipeptide conformations ($\mathbb{R}^{21}$)

RMSD distance (non-Euclidean)



Note: Spectral Clustering takes a week of tweaking, while ToMATo runs out-of-the-box in a few minutes

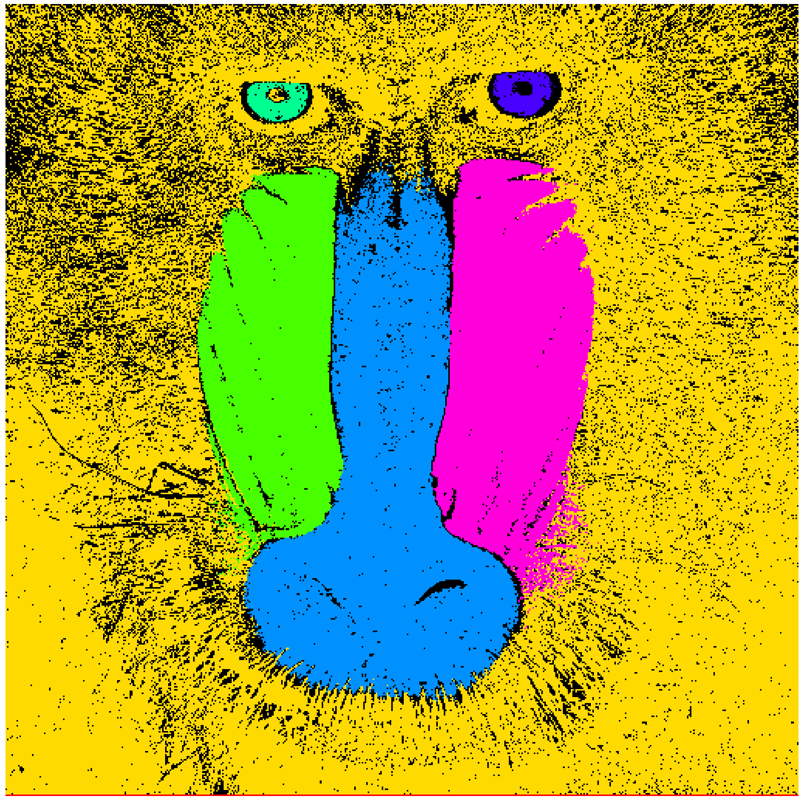
• Y. Yao, J. Sun, X. Huang, G. Bowman, G. Singh, M. Lesnick, L. Guibas, V. Pande, G. Carlsson, Topological methods for exploring low-density states in biomolecular folding pathways, *The Journal of Chemical Physics*, 2009.

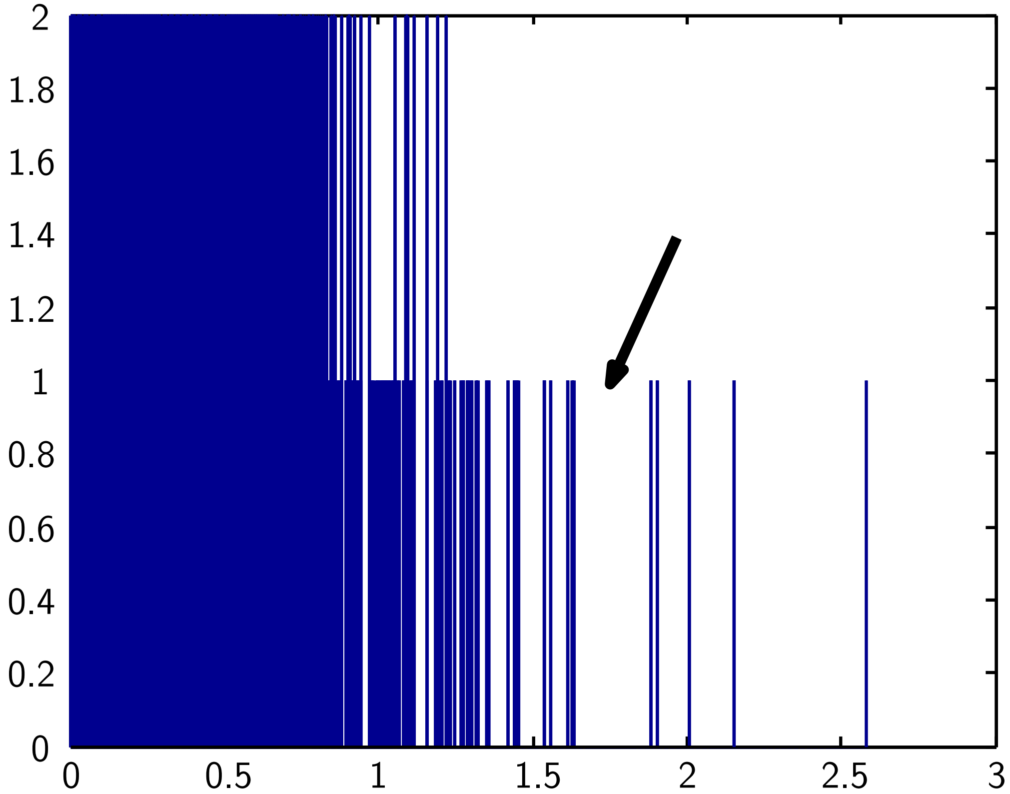
Experimental Results

Image Segmentation

Density is estimated in 3D color space (Luv)

Neighborhood graph is built in image domain





Distribution of prominences does not usually show a clear unique gap

Still, relationship between choice of τ and number of obtained clusters remains explicit

Recap'

ToMATo:

1. graph-based mode-seeking algorithm of [KNF'76]
2. single-pass cluster merging phase guided by persistence

Competitors:

1. Mean-Shift and its variants (smoothing a priori)
2. ...

Recap'

- Highly generic
  - applicable in arbitrary metric spaces
  - agnostic to the choice of neighborhood graph and density estimator
- Easy to tune
  - mostly two parameters: neighborhood size, persistence threshold τ
  - PD provides insight into the correct number of clusters
- Comes with theoretical guarantees
  - number of obtained clusters versus number of prominent peaks
  - partial approximation of the basins of attraction of the peaks
- Efficient and practical
  - near linear runtime, linear main memory usage
  - can handle data sets with hundreds of thousands of points in practice