MPRI – Computation Geometry and Topology

Clustering

Steve Oudot

(steve.oudot@inria.fr)

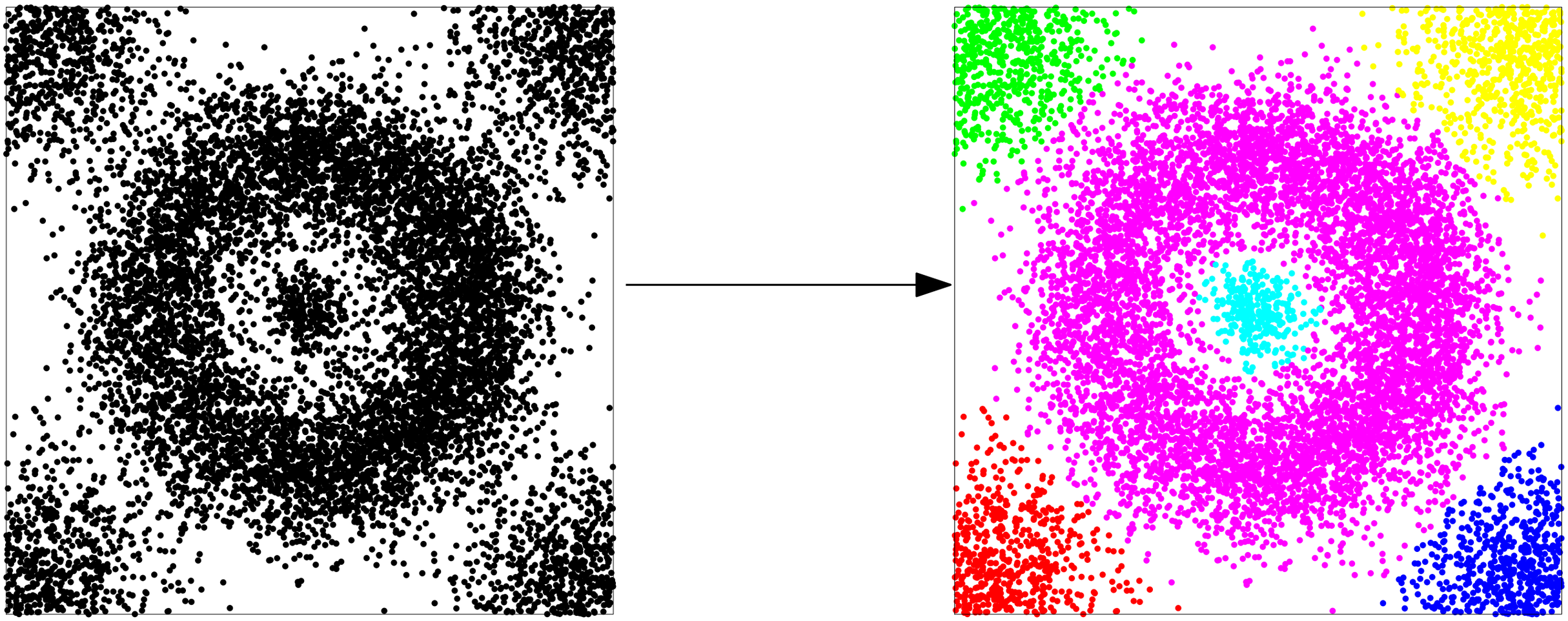
Inria

ÉCOLE POLYTECHNIQUE

Cluster Analysis

Input: a finite set of observations: - point cloud with coordinates

- distance / (dis-)similarity matrix



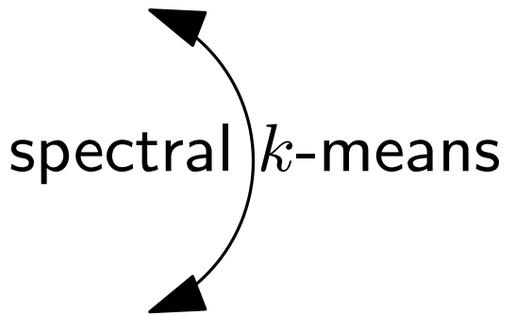
Task:

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

A Wealth of Approaches

Variational

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



Spectral

- 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

Density thresholding

- DBSCAN
- OPTICS

Mode seeking

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

Hierarchical divisive/agglomerative

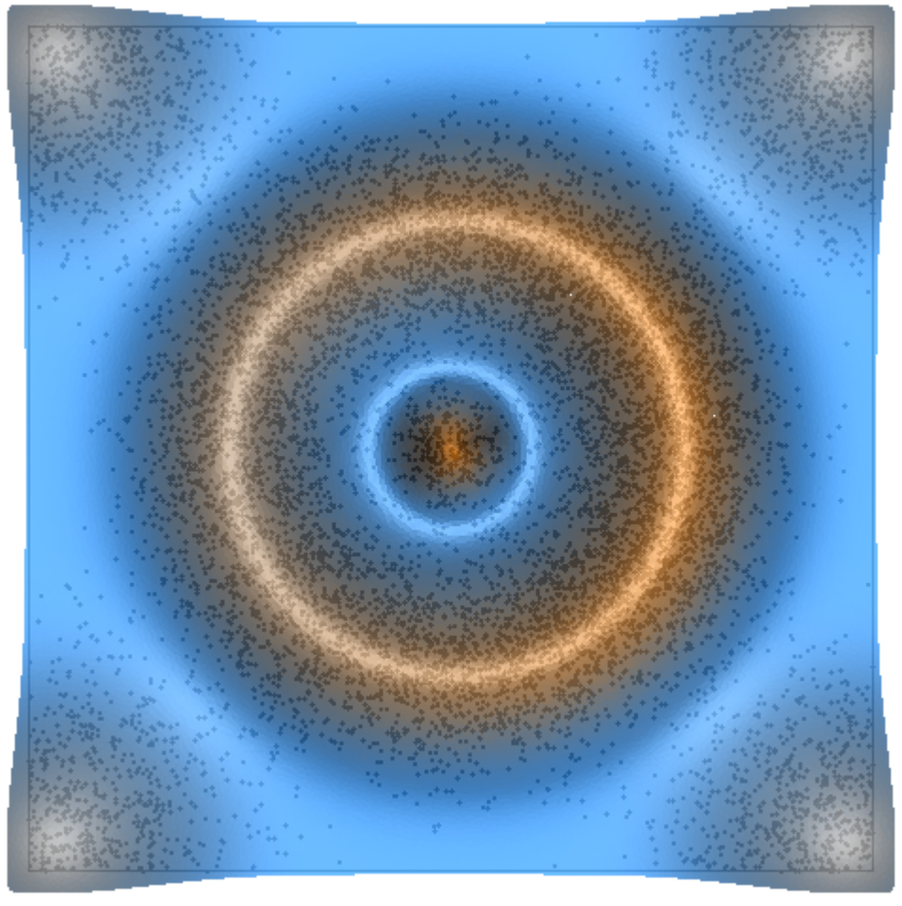
- single-linkage
- BIRCH

Valley seeking

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

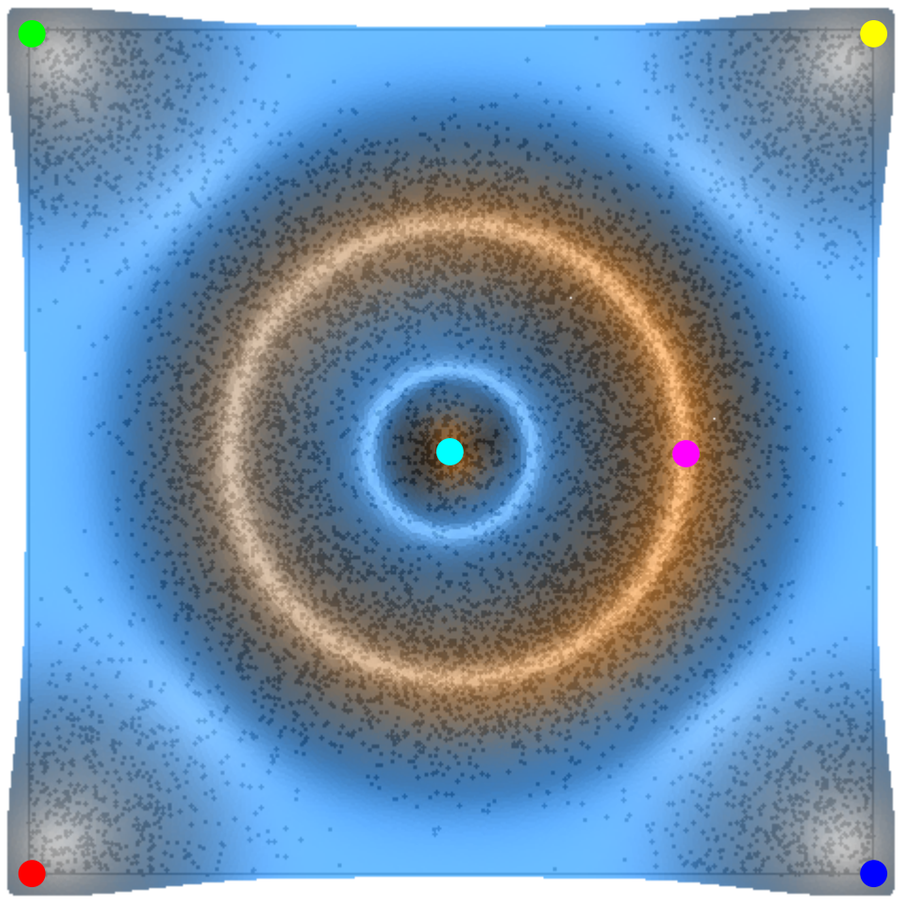
Mode-Seeking Paradigm

- 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



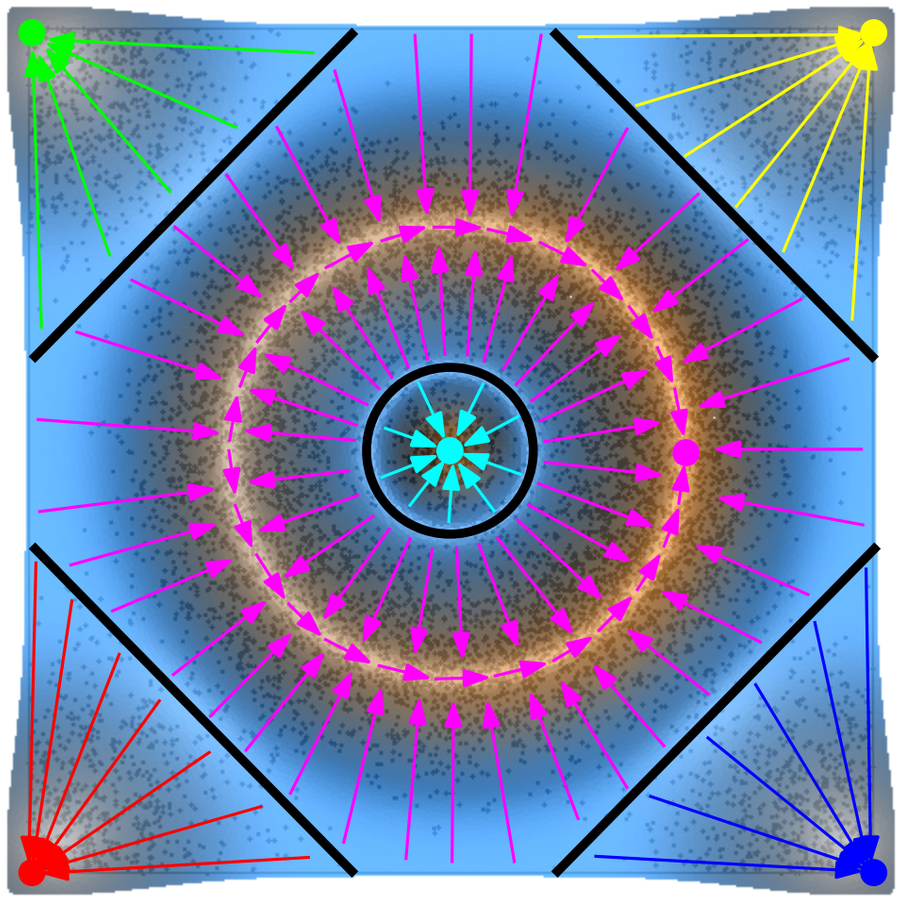
Mode-Seeking Paradigm

- 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



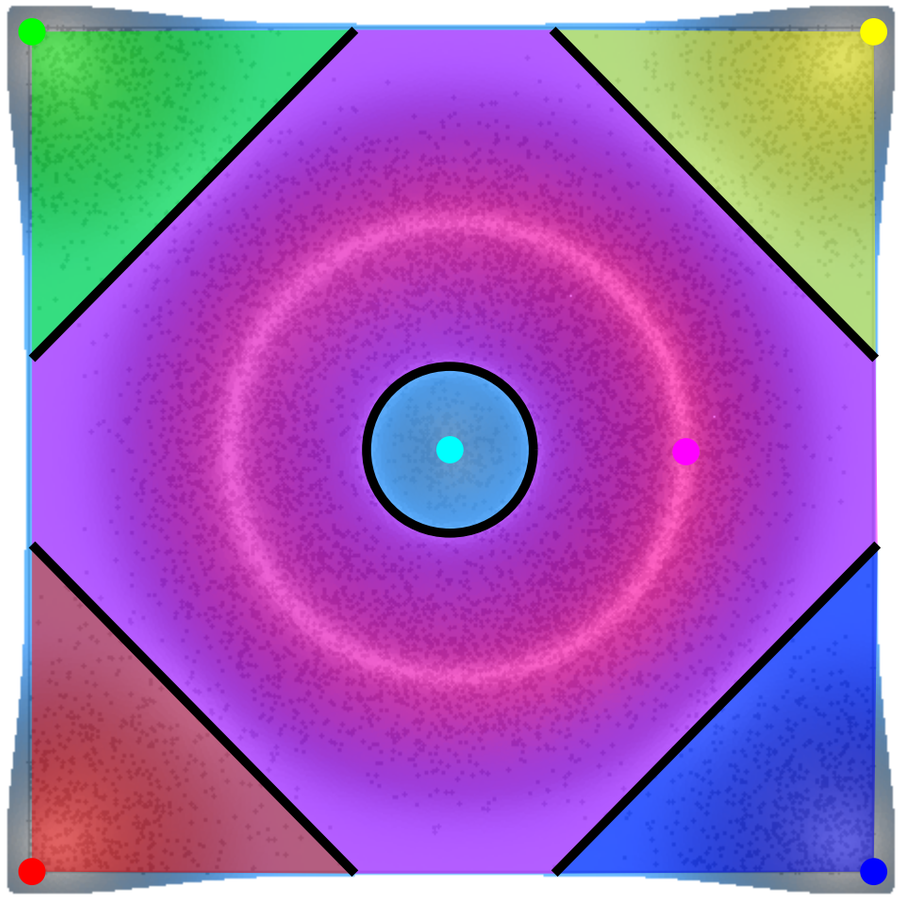
Mode-Seeking Paradigm

- 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



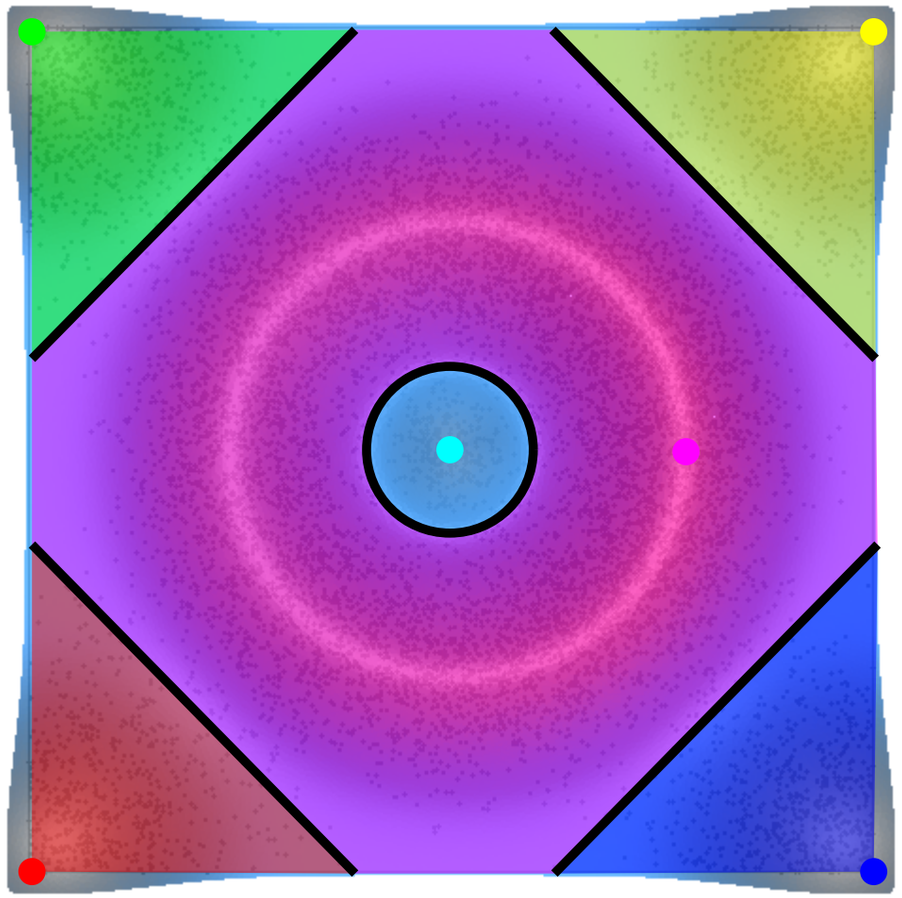
Mode-Seeking Paradigm

- 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



Approaches to mode seeking

• **Numerical:** push data points along approximate flow lines of the underlying density function in ambient space

D. Comaniciu, P. Meer (2002): Mean shift: A robust approach toward feature space analysis, IEEE Trans. on Pattern Analysis and Machine Intelligence, *24(5):603–619.*

• **Combinatorial:** approximate gradient flow lines globally in some neighrborhood graph, clusters given by trees in a spanning forest

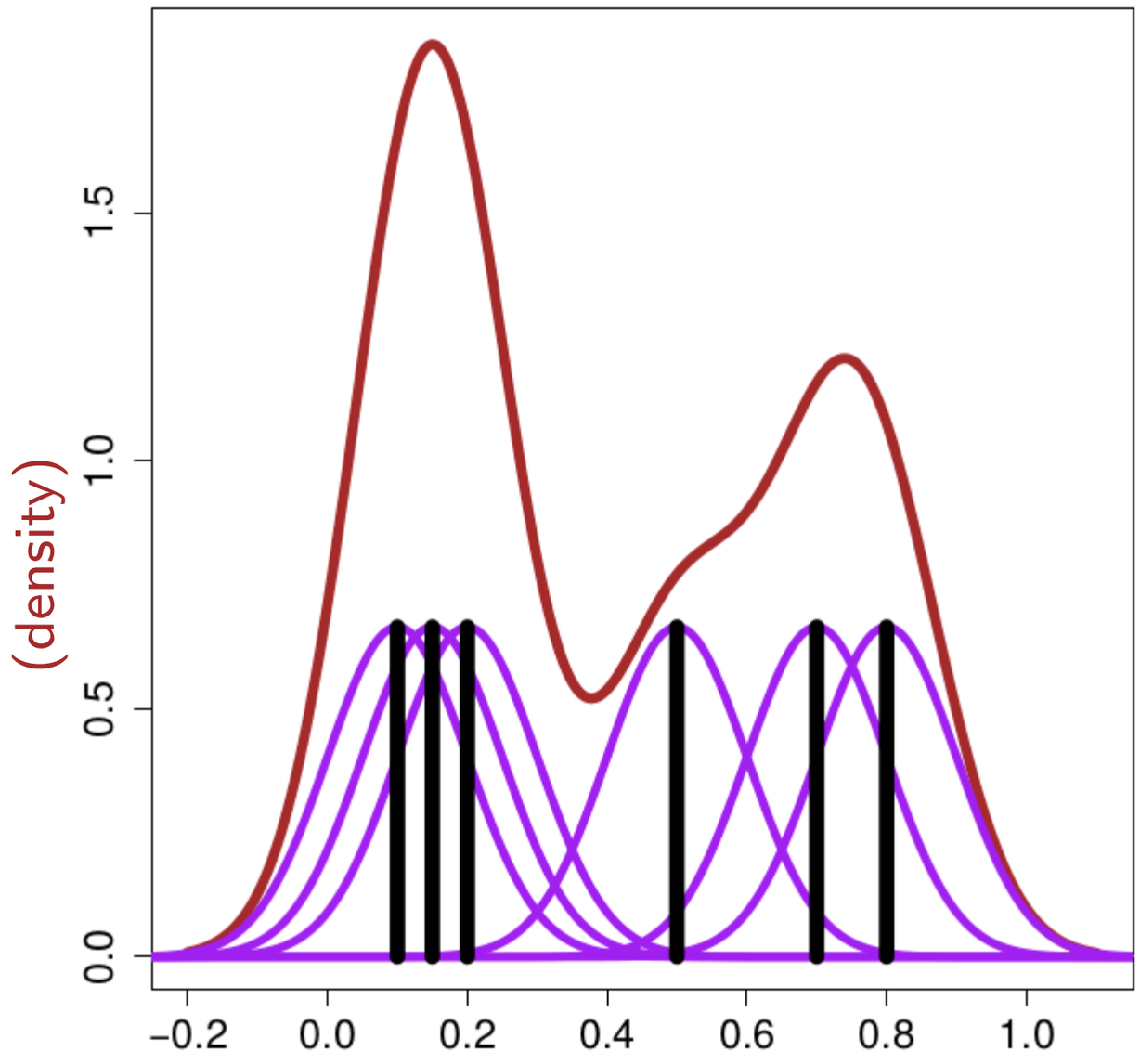
W. L. Koontz, P. M. Narendra, K. Fukunaga [1976]: A graph-theoretic approach to nonparametric cluster analysis. IEEE Trans. on Computers, *24:936–944.*

F. Chazal, L. Guibas, S. Oudot, P. Skraba (2013): ToMATo: Persistence-based clustering in Riemannian manifolds, Journal of the ACM, *volume 60, issue 6, article 41.*

1. Mean-Shift

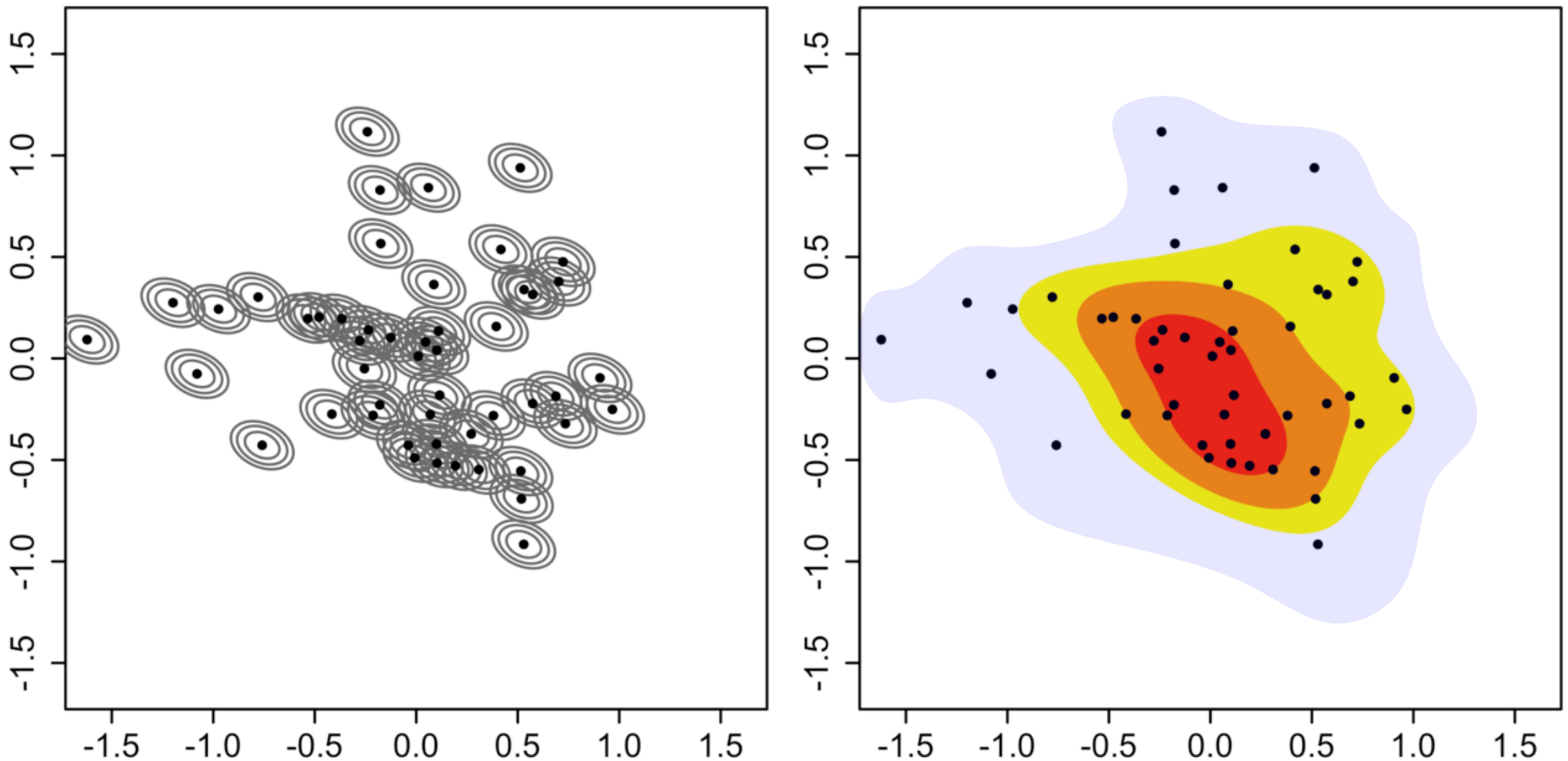
Kernel density estimators

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:

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

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

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

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

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

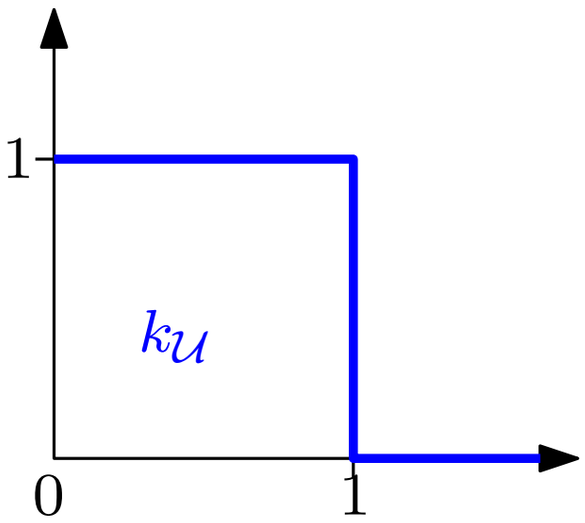
$$\rightsquigarrow \quad \hat{f}_{\sigma,k}(x) := \frac{c_{k,d}}{n \, \sigma^d} \sum_{i=1}^{n} k\left(\frac{\|x - p_i\|_2^2}{\sigma^2} \right)$$

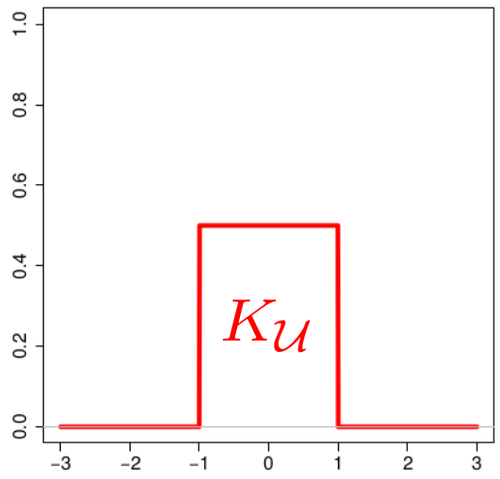
Common kernels

Flat / Uniform: $k_{\mathcal{U}}(t) := \begin{cases} 1 \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$

$$\rightsquigarrow c_{k,d} = 1/\mathrm{Vol}\, B_d(0,1)$$

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

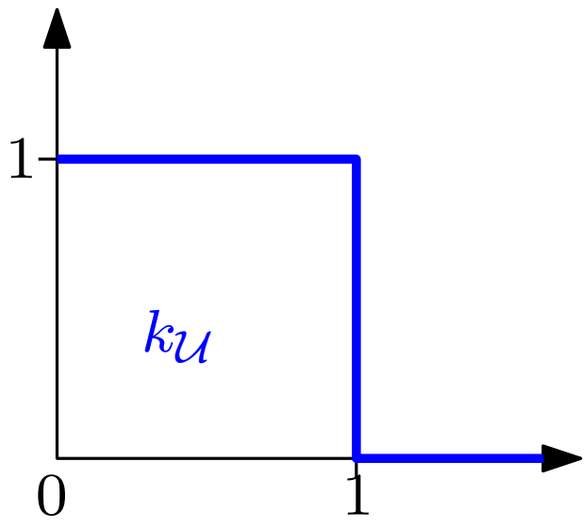


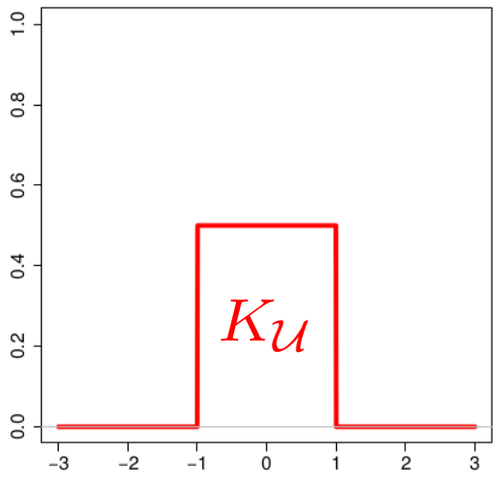


Common kernels

Flat / Uniform: $k_{\mathcal{U}}(t) := \begin{cases} 1 \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$

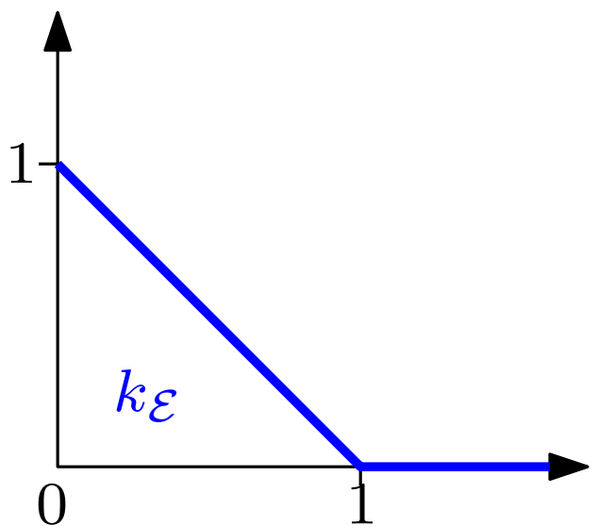
$$\rightsquigarrow c_{k,d} = 1/\operatorname{Vol} B_d(0,1) = \frac{\Gamma(d/2+1)}{\pi^{d/2}}$$

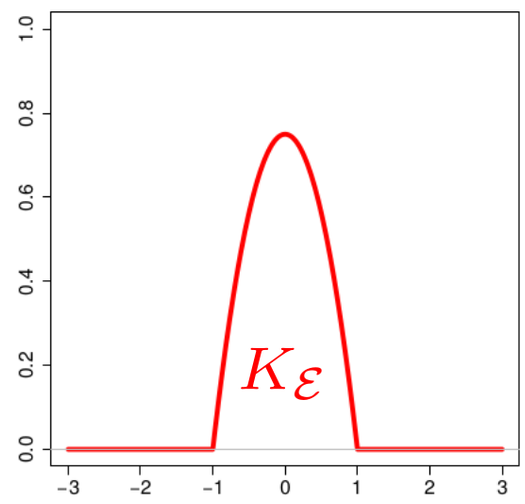




Epanechnikov: $k_{\mathcal{E}}(t) := \begin{cases} 1-t \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$

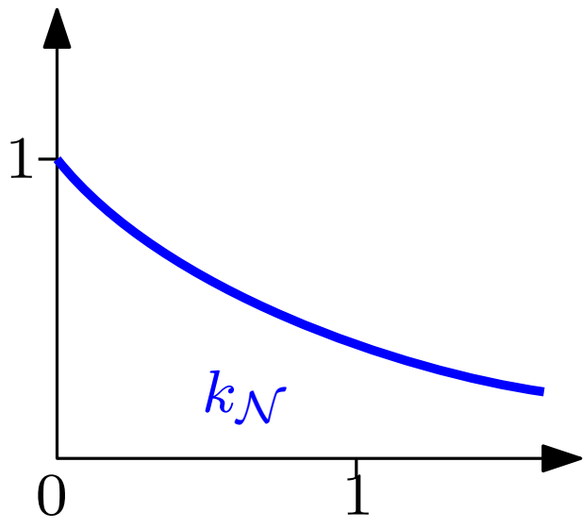
$$\rightsquigarrow c_{k,d} = \frac{d+2}{2\operatorname{Vol} B_d(0,1)}$$

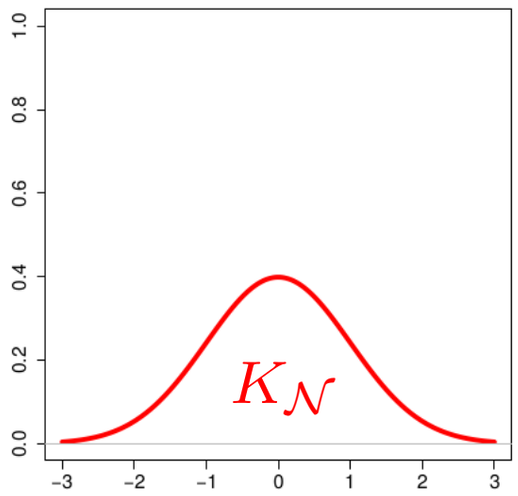




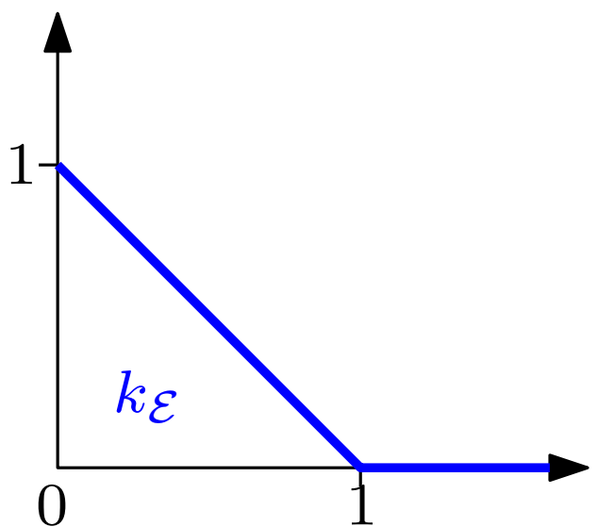
Common kernels

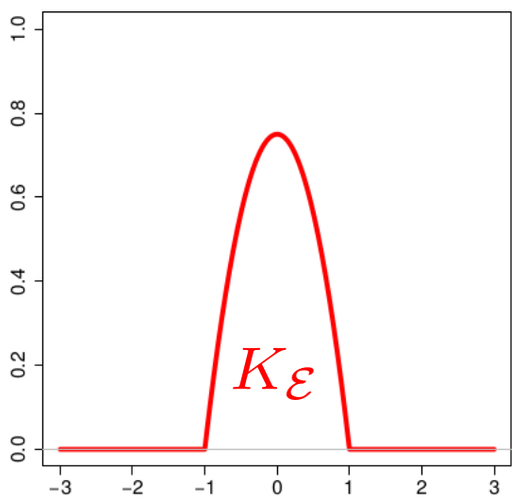
Gaussian: $k_{\mathcal{N}}(t) := \exp(-t/2)$ $\rightsquigarrow c_{k,d} = (2\pi)^{-d/2}$



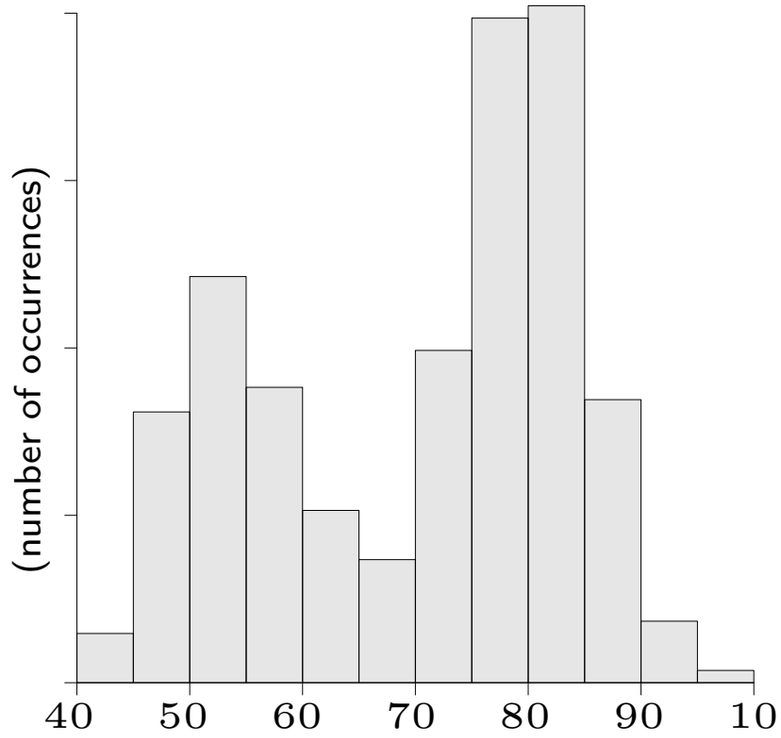


Epanechnikov: $k_{\mathcal{E}}(t) := \begin{cases} 1-t \text{ if } t \leq 1 \\ 0 \text{ if } t > 1 \end{cases}$ $\rightsquigarrow c_{k,d} = \dfrac{d+2}{2\,\mathrm{Vol}\,B_d(0,1)}$



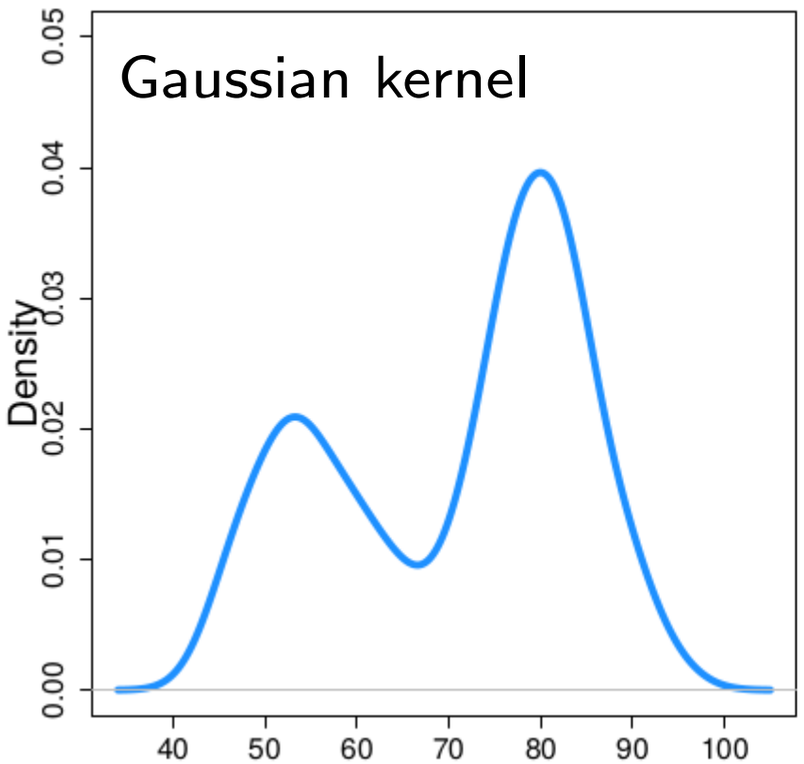


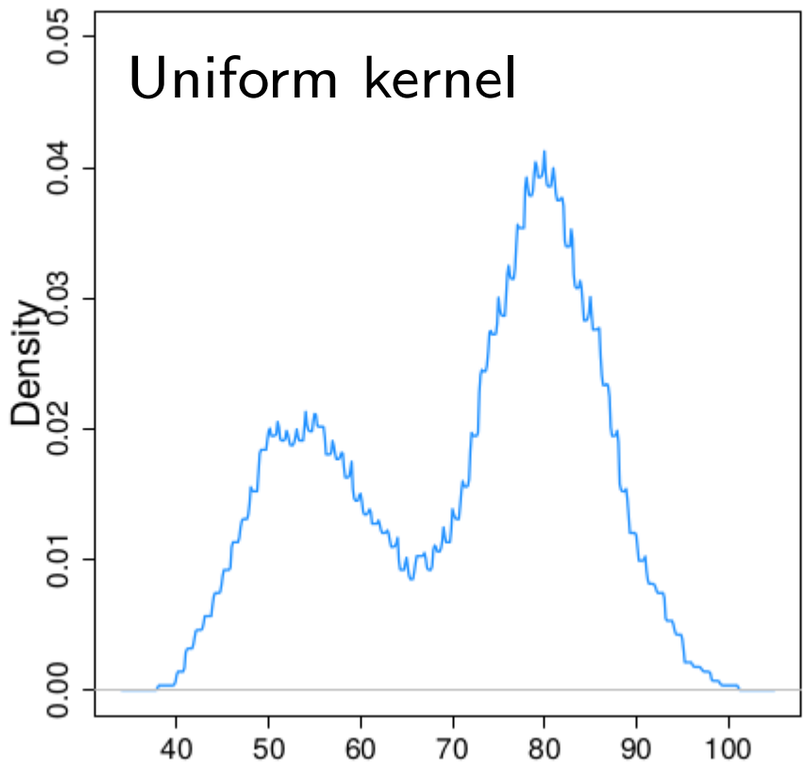
Common kernels

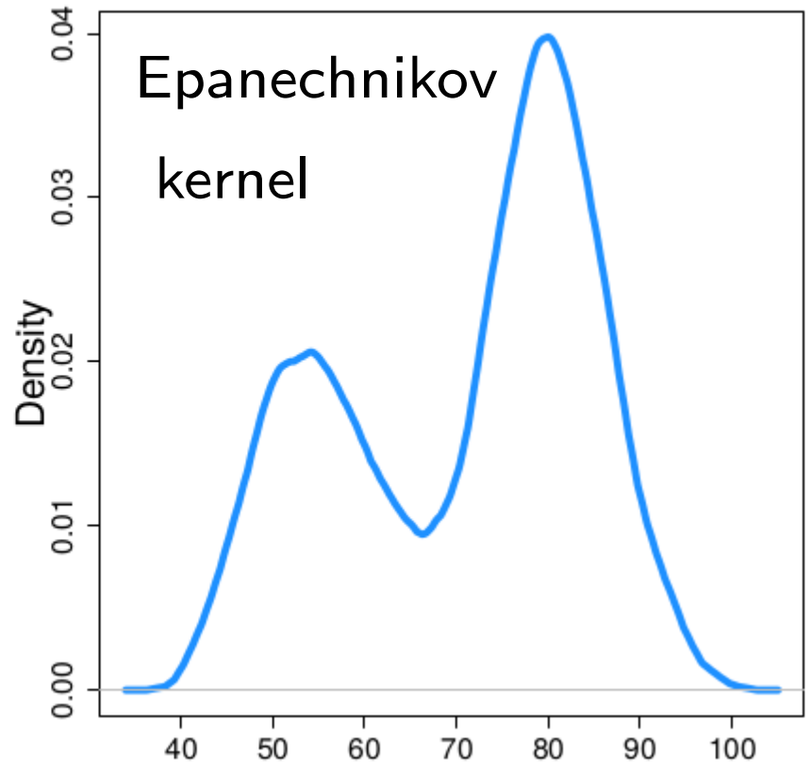


Old faithful geyser dataset (available in R):

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

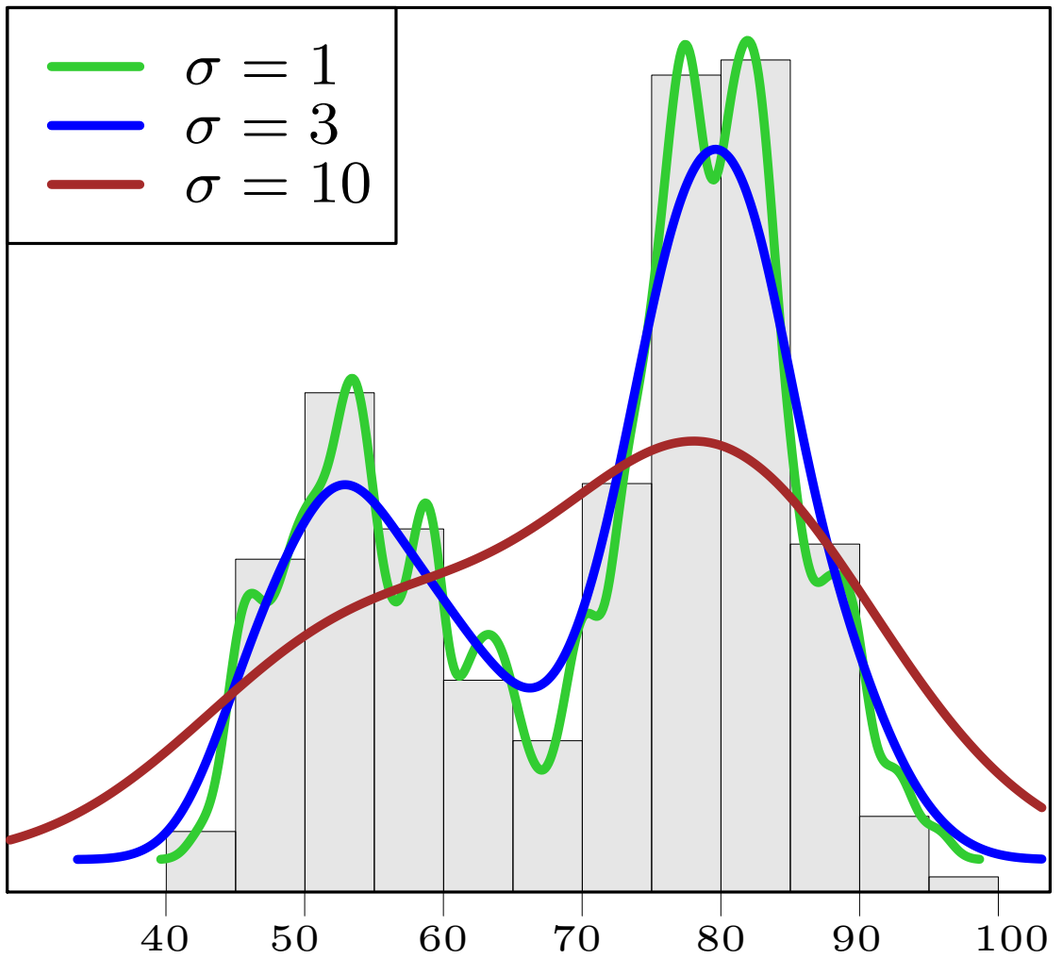






Influence of the bandwidth

- 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):

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

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\textbf{mean-shift } m_{\sigma,g}(x)}$$

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):

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

$$\underbrace{\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad}_{\textbf{mean-shift } m_{\sigma,g}(x)}$$

$\Rightarrow$ 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)

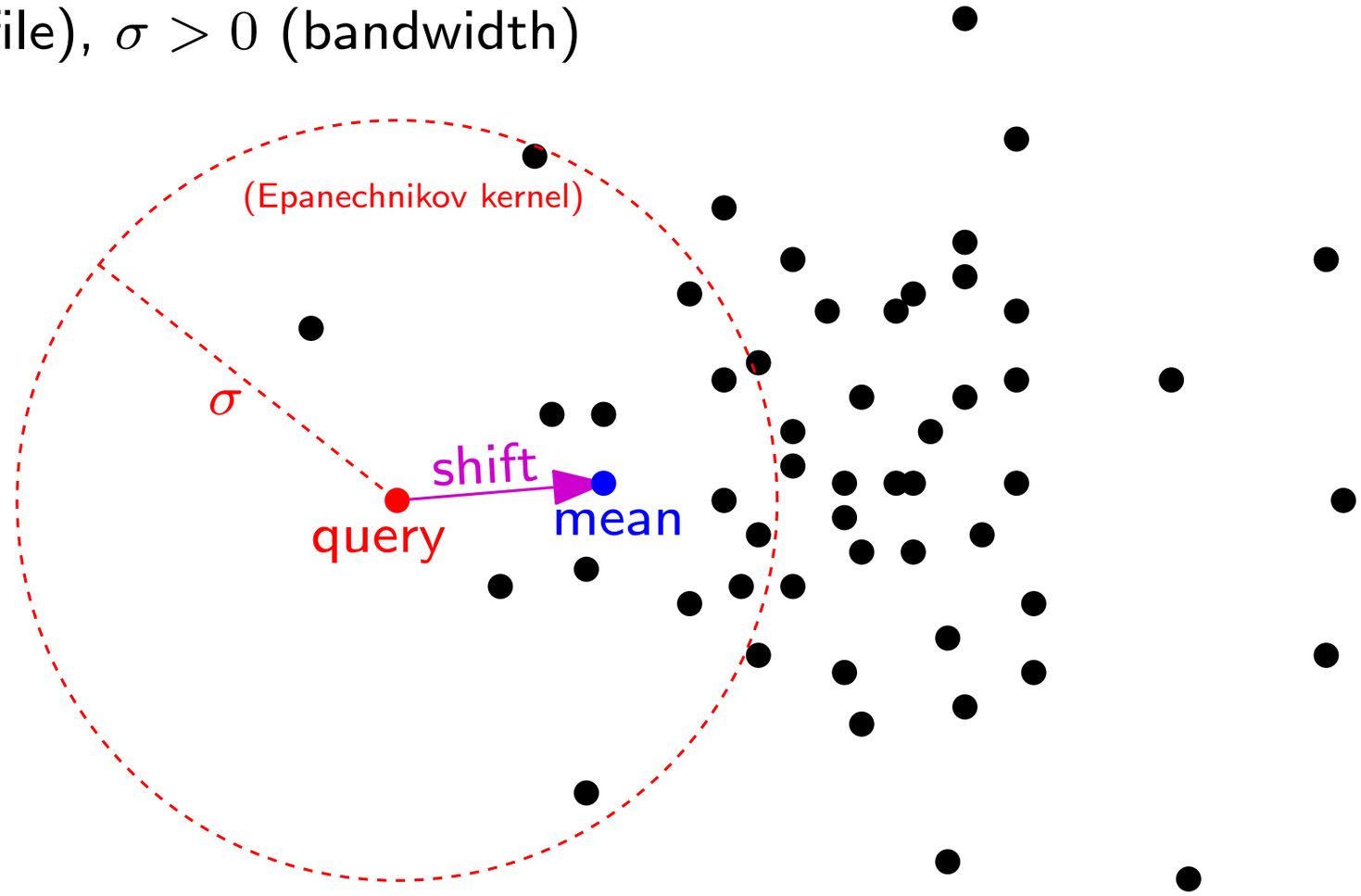
Parameters: $k : \mathbb{R}^+ \to \mathbb{R}^+$ (profile), $\sigma > 0$ (bandwidth)

$x_0 := x$

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)

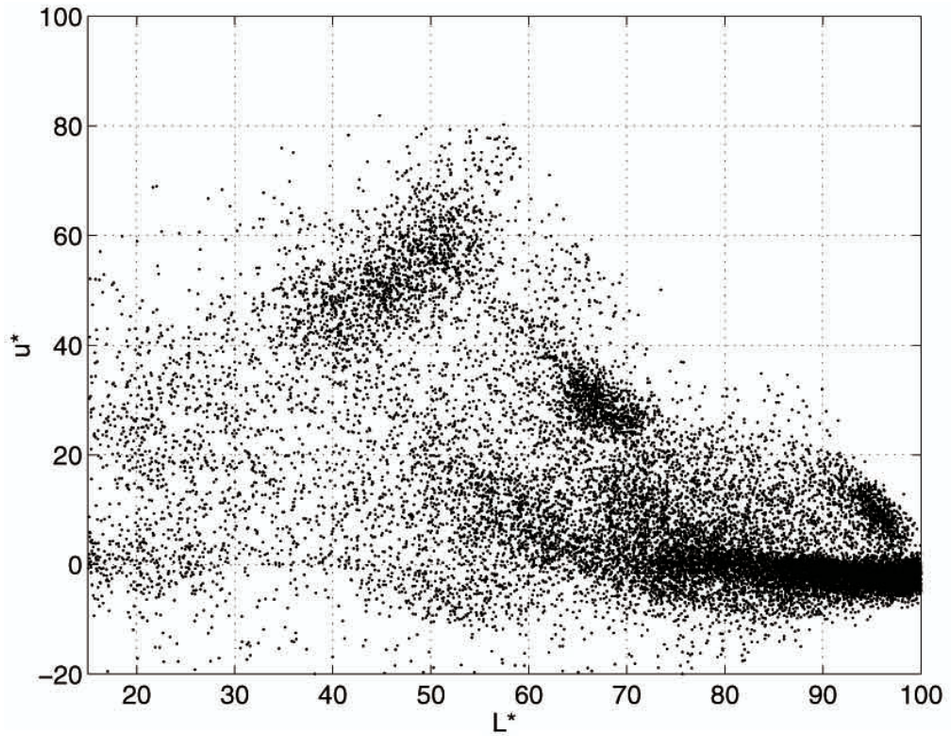
  $\rightarrow$ identification of modes (mode radius)

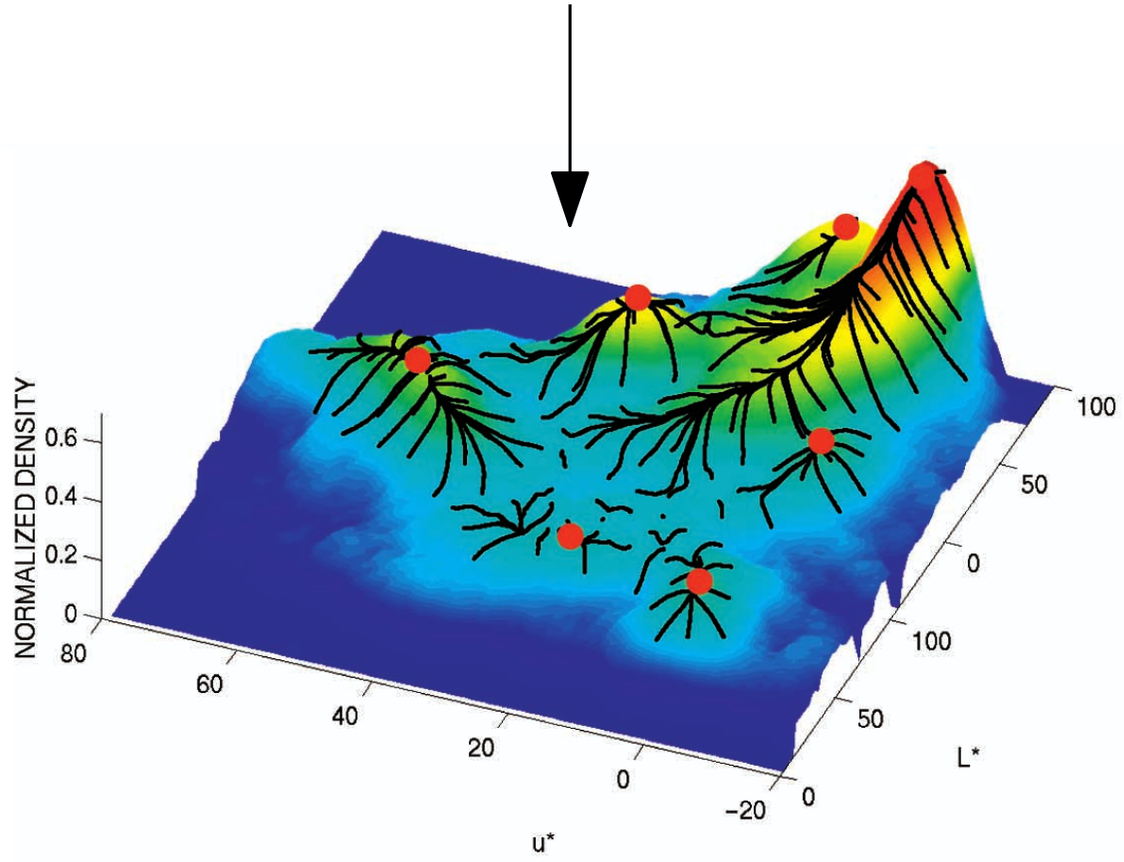
  $\rightarrow$ speed-up: hill-climbing gathers neighboring points (gathering radius)

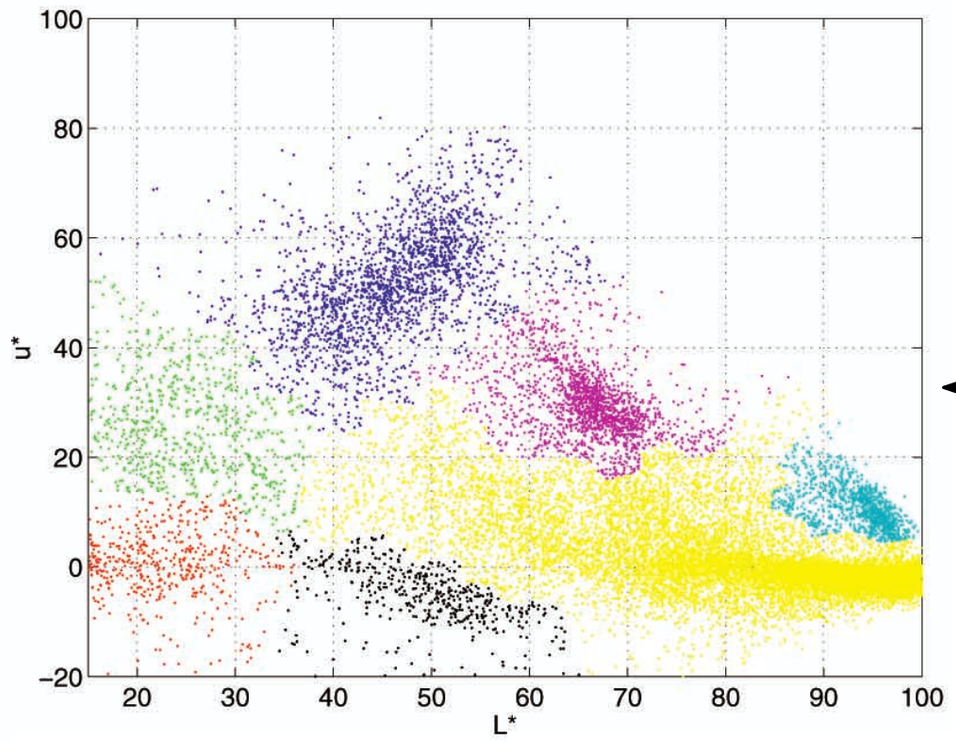
$\rightsquigarrow$ heuristic: make these radii proportional to the estimator's bandwidth σ

Examples [Comaniciu, Meer 2002]









Examples [Comaniciu, Meer 2002]



Original



$(h_s, h_r) = (8, 8)$



$(h_s, h_r) = (8, 16)$



$(h_s, h_r) = (16, 4)$



$(h_s, h_r) = (16, 8)$



$(h_s, h_r) = (16, 16)$



$(h_s, h_r) = (32, 4)$



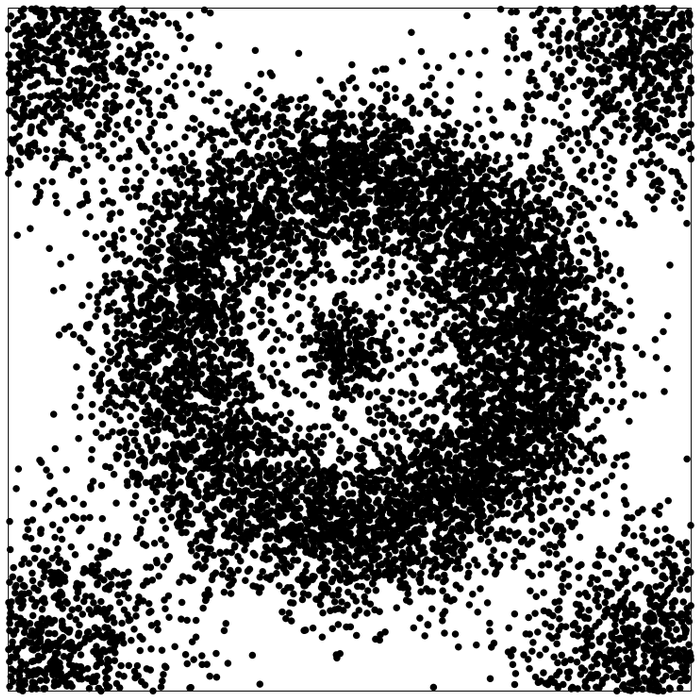
$(h_s, h_r) = (32, 8)$



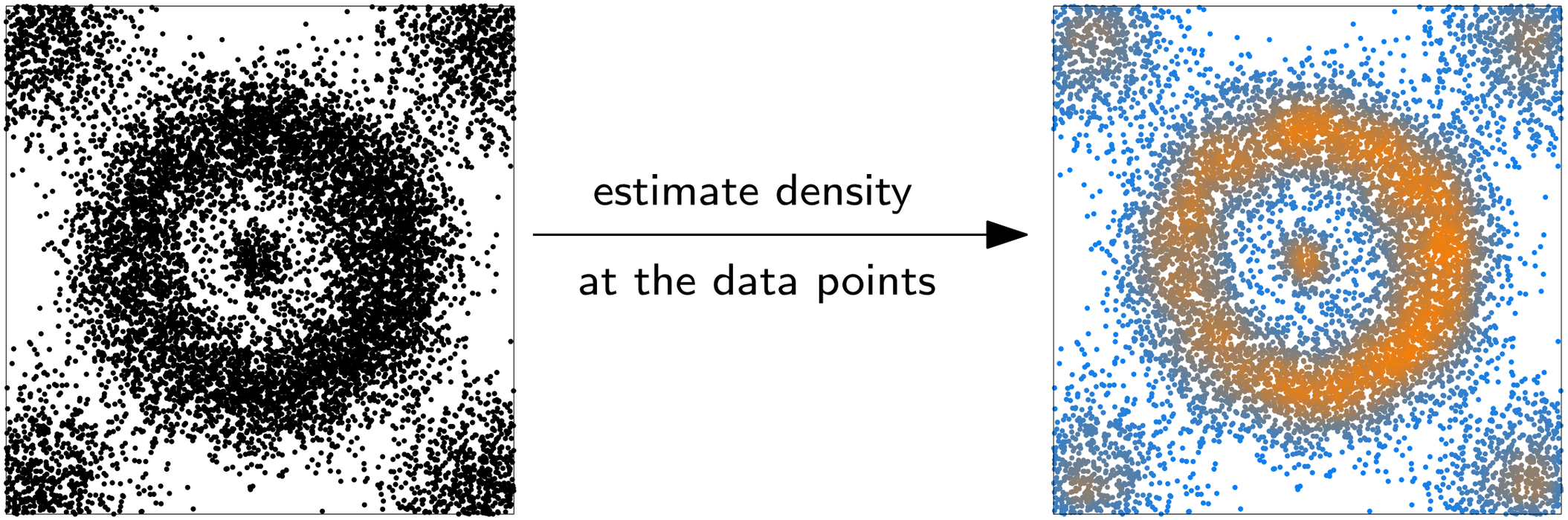
$(h_s, h_r) = (32, 16)$

2. Graph-based hill-climbing

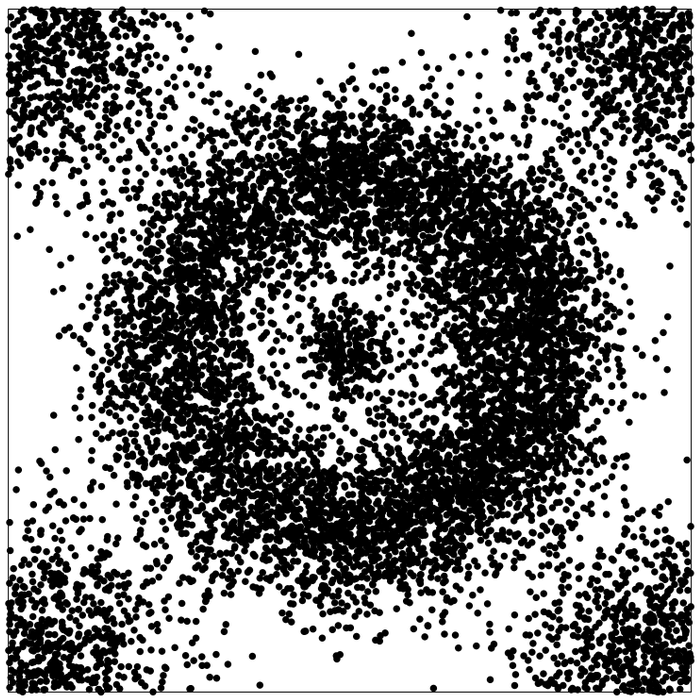
[Koontz, Narendra, Fukunaga'76] in a Nutshell



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

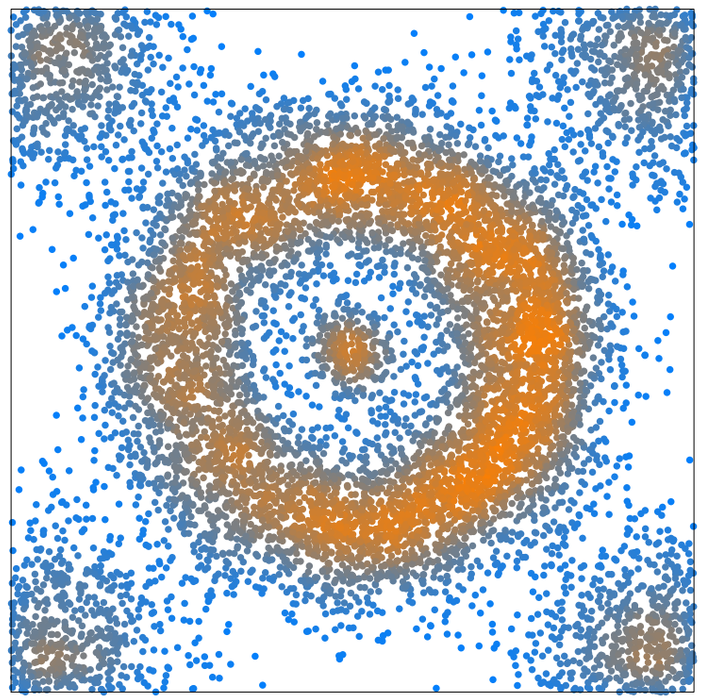


[Koontz, Narendra, Fukunaga'76] in a Nutshell

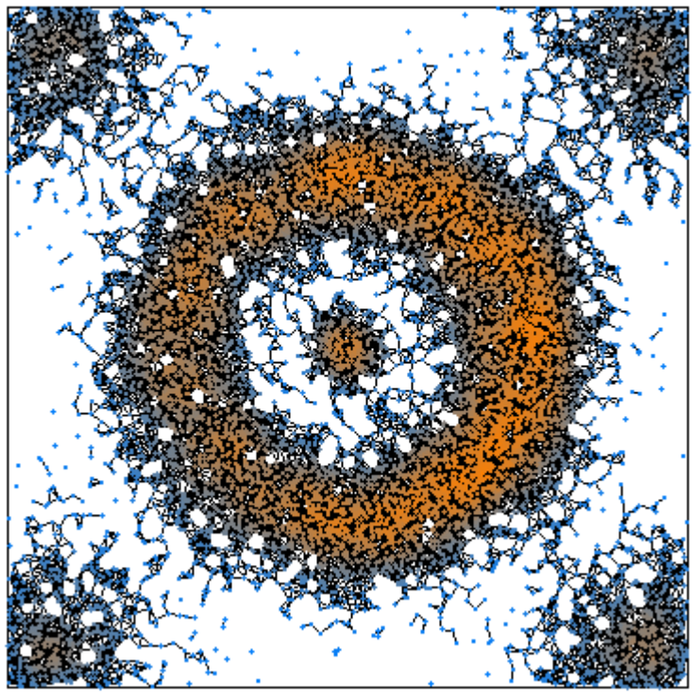


estimate density

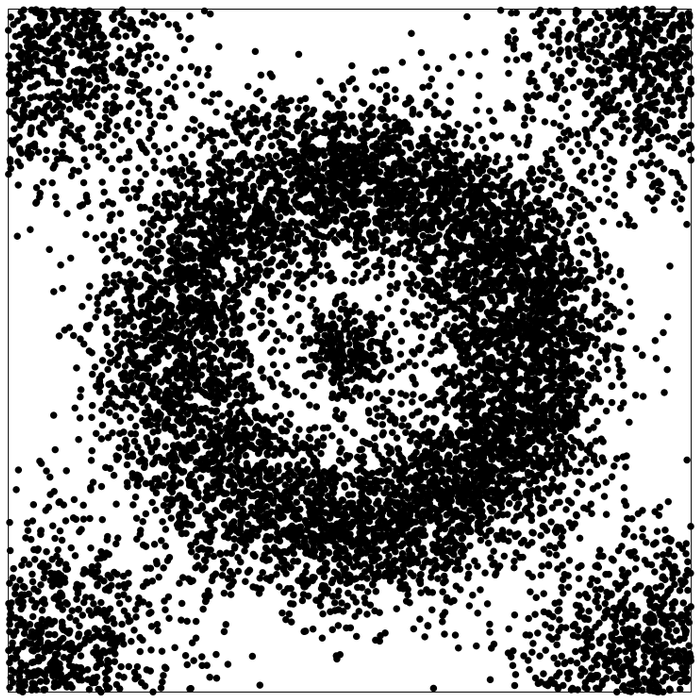
at the data points



build neighborhood graph

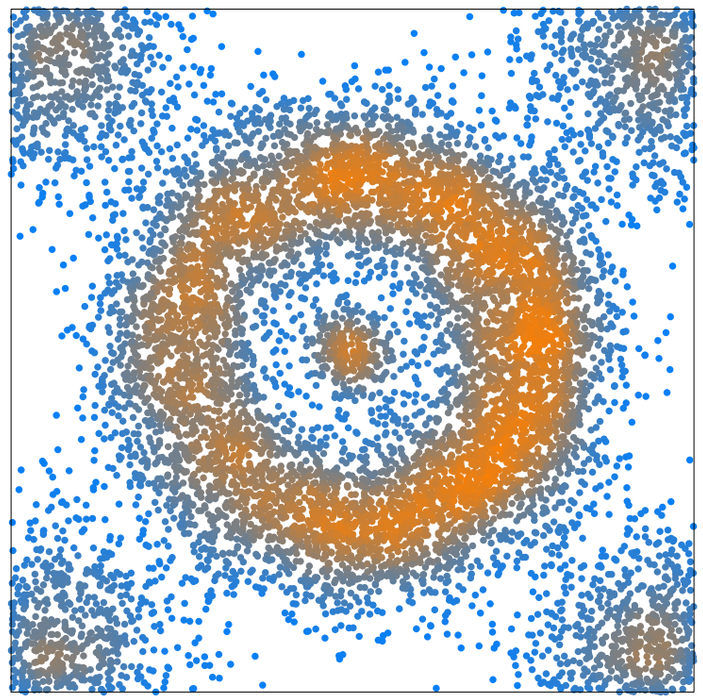


[Koontz, Narendra, Fukunaga'76] in a Nutshell

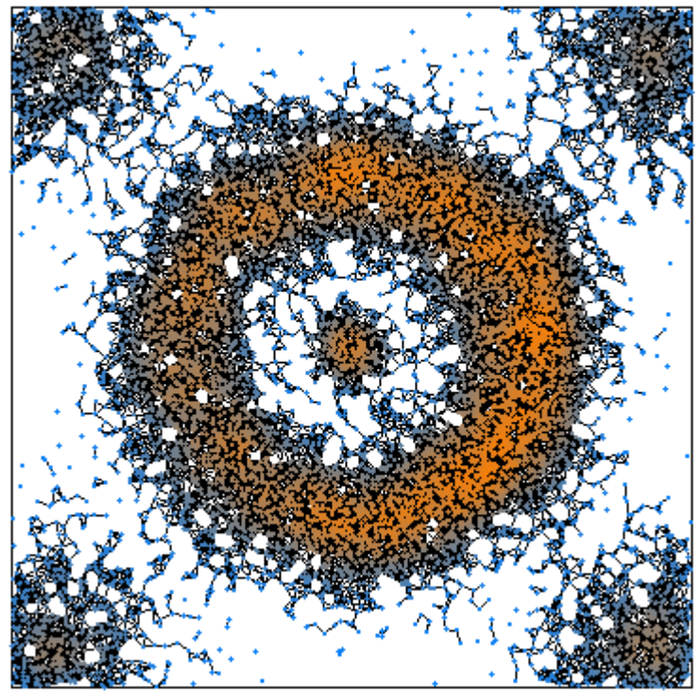


estimate density

at the data points

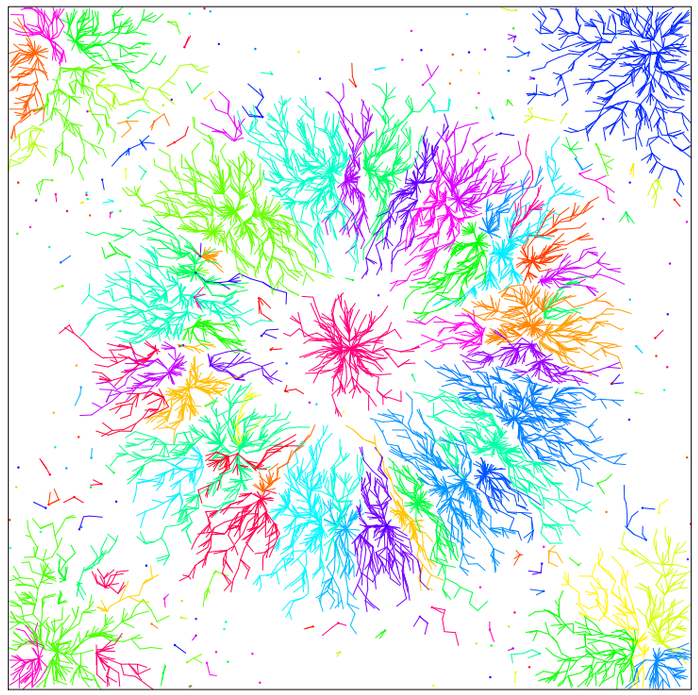


build neighborhood graph



approximate gradient

by a graph edge
at each data point



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;

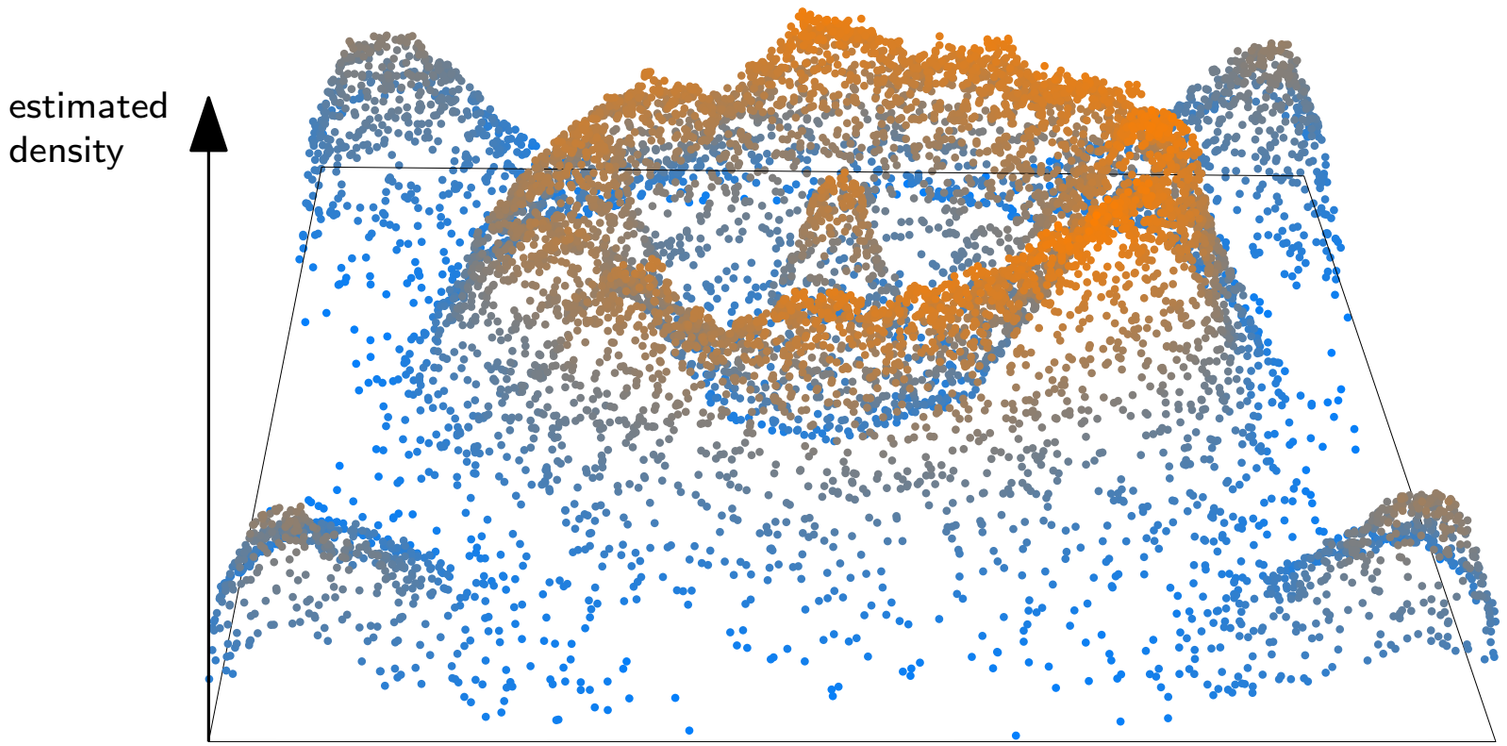
for $i = 1$ to n **do**
    Let $\mathcal{N}$ be the set of neighbors of i in G that have indices lower than i;
    if $\mathcal{N} = \emptyset$ *// vertex i is a peak of $\hat{f}$ within G*
        Create a new entry e in $\mathcal{U}$ and attach vertex i to it;
        $r(e) \leftarrow i$ *// $r(e)$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

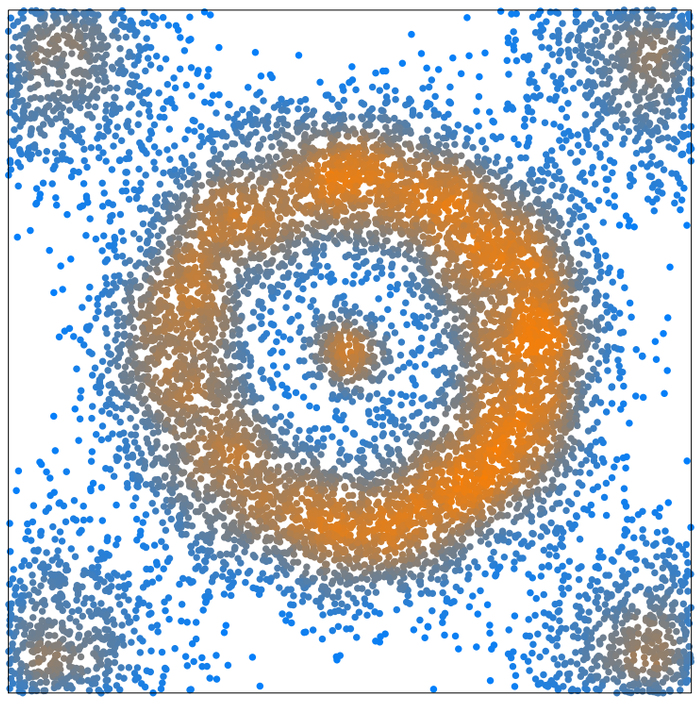
graph-based hill-climbing (1976)

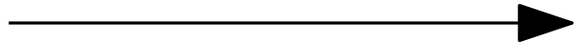
Output: the collection of entries e in $\mathcal{U}$

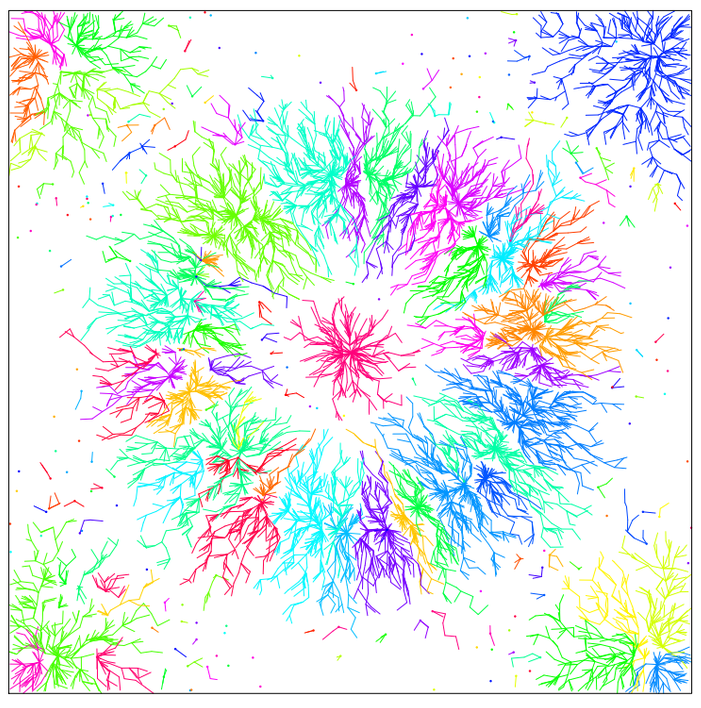
Why things go ill

- Noisy estimator



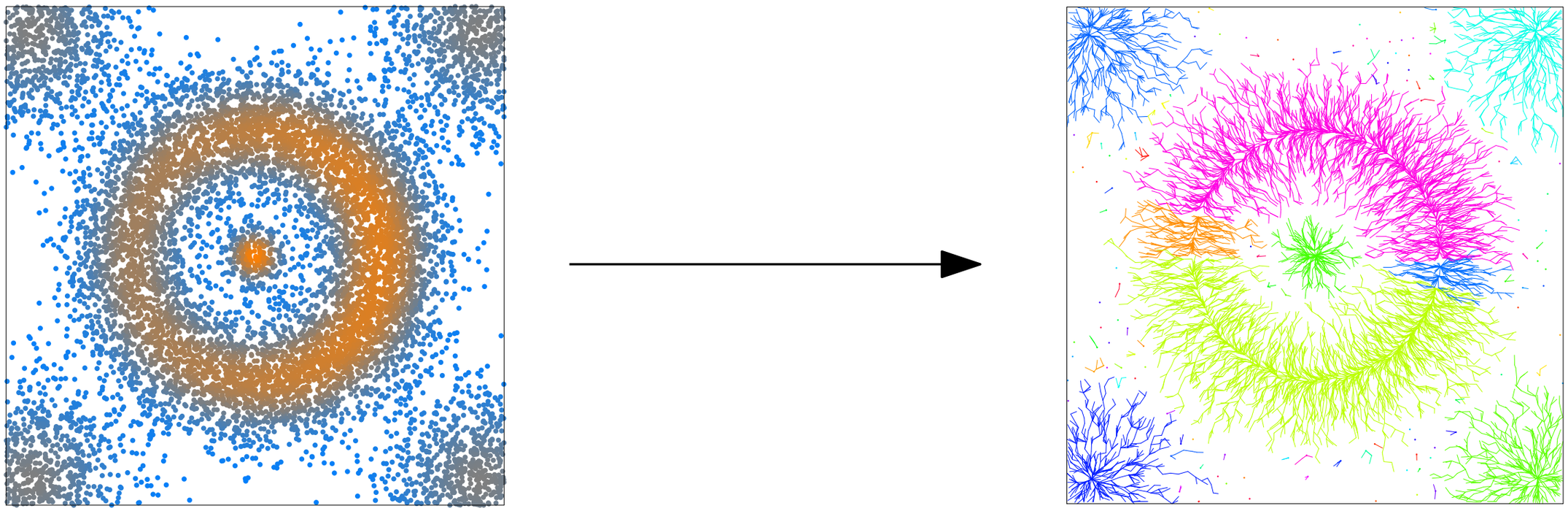






Why things go ill

- Noisy estimator
- Neighborhood graph



Why things go ill

- Noisy estimator
- Neighborhood graph

Fix: merge clusters after clustering

use topological persistence theory to guide a single-pass merging step:

→ estimate prominences of peaks

→ merge clusters of low-prominence peaks into clusters of high-prominence peaks

F. Chazal, L. Guibas, S. Oudot, P. Skraba (2013): ToMATo: Persistence-based clustering in Riemannian manifolds, Journal of the ACM, *volume 60, issue 6, article 41.*

Prominence / Persistence of the peaks of a real-valued function

(a.k.a. persistence theory in degree 0)

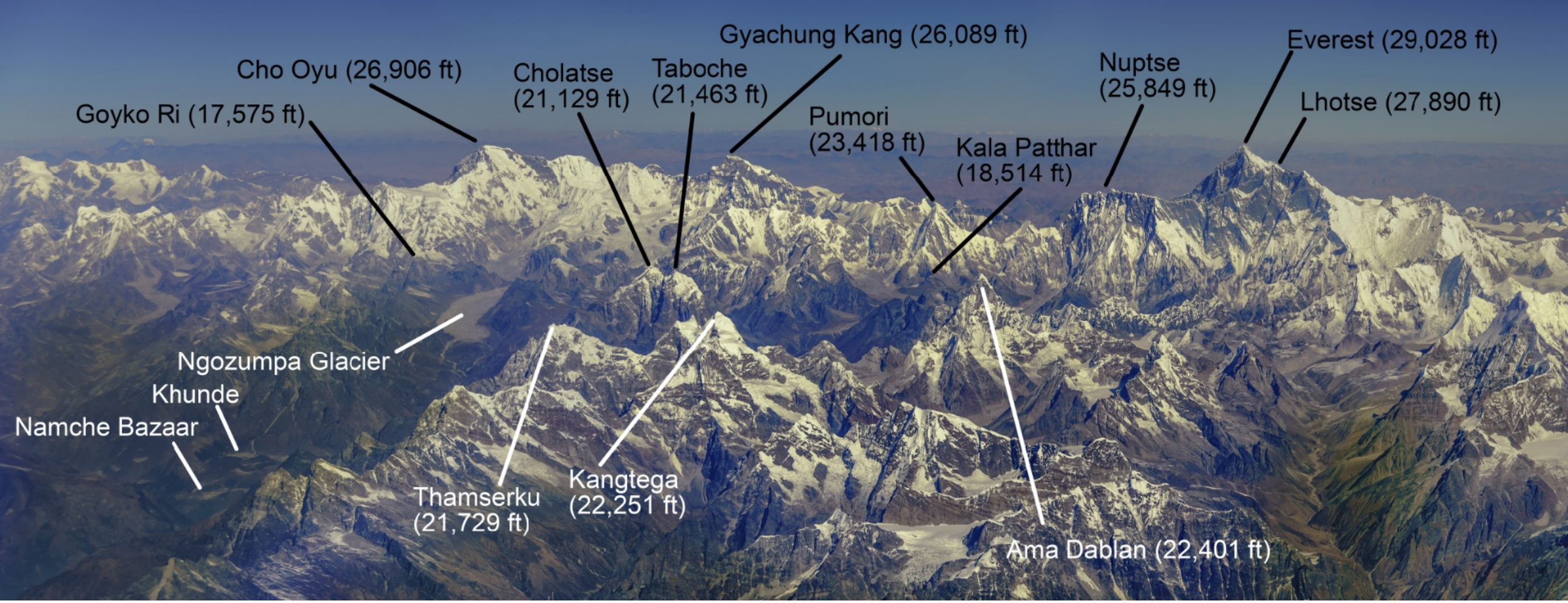


photo credits: Jim Block (http://www.gigapan.com/gigapans/198046/)

summit	continent	**altitude** (m)
Mount Everest	Asia	8849
K2	Asia	8611
Kangchenjunga	Asia	8586
Lhotse	Asia	8516
Makalu	Asia	8485
Cho Oyu	Asia	8188
Dhaulagiri I	Asia	8167
Manaslu	Asia	8163
Nanga Parbat	Asia	8126
Annapurna I	Asia	8091

source: Wikipedia (https://en.wikipedia.org/wiki/List_of_highest_mountains_on_Earth)

prominence threshold : 500m

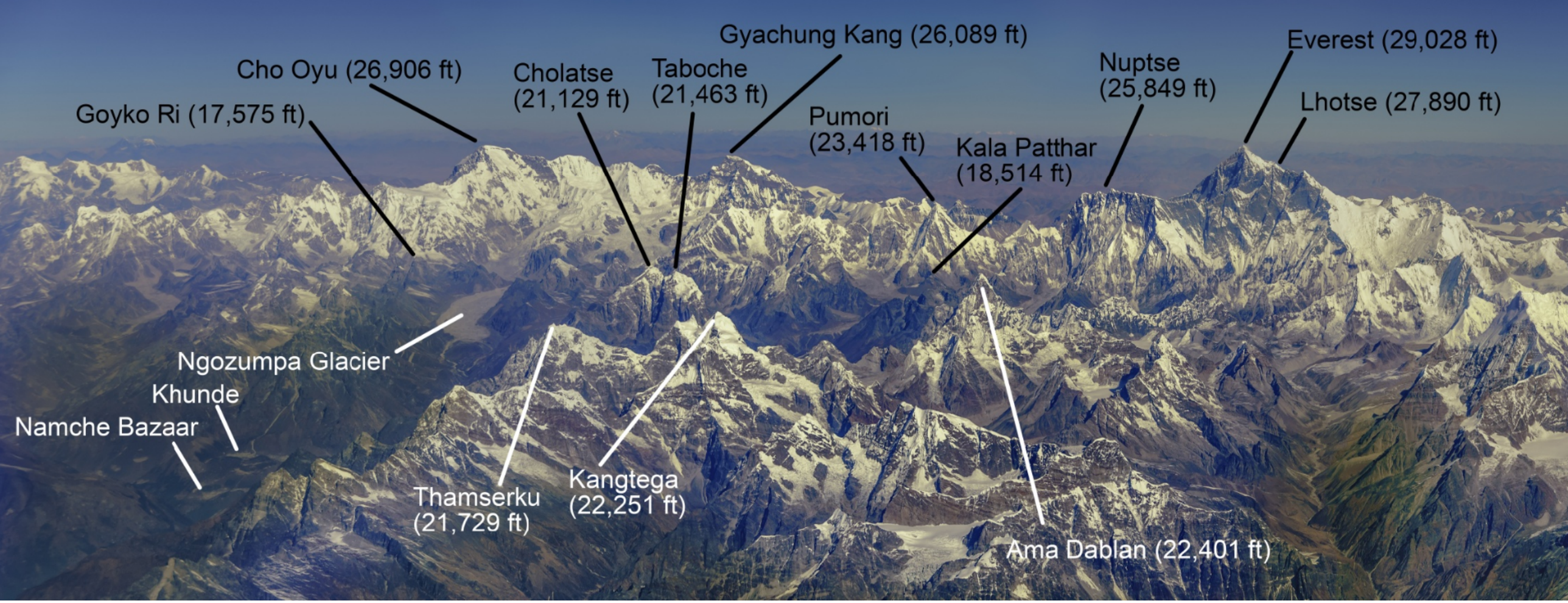


photo credits: Jim Block (http://www.gigapan.com/gigapans/198046/)

summit	continent	**altitude** (m)	prominence (m)
Mount Everest	Asia	8849	8849
K2	Asia	8611	4020
Kangchenjunga	Asia	8586	3922
Lhotse	Asia	8516	610
Makalu	Asia	8485	2378
Cho Oyu	Asia	8188	2340
Dhaulagiri I	Asia	8167	3357
Manaslu	Asia	8163	3092
Nanga Parbat	Asia	8126	4608
Annapurna I	Asia	8091	2984

prominence threshold : 500m

source: Wikipedia (https://en.wikipedia.org/wiki/List_of_highest_mountains_on_Earth)

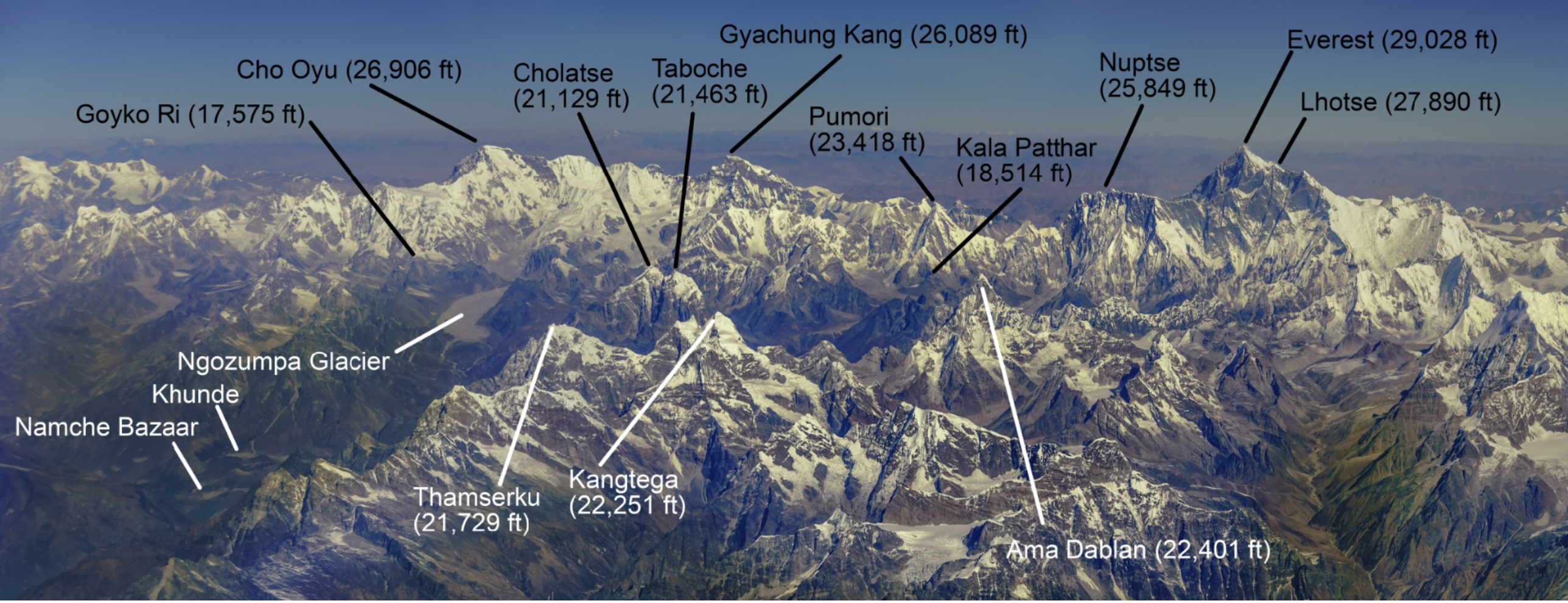
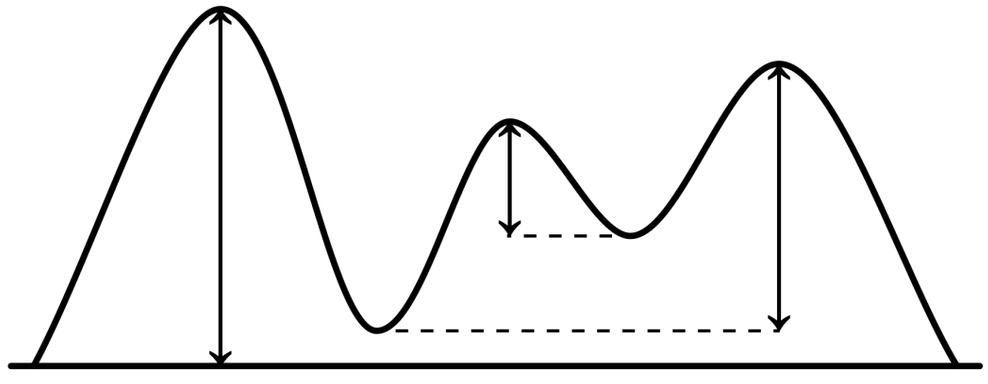


photo credits: Jim Block (http://www.gigapan.com/gigapans/198046/)

prominence of a peak
(according to alpinists):

difference in altitude with the highest pass allowing to reach a higher peak

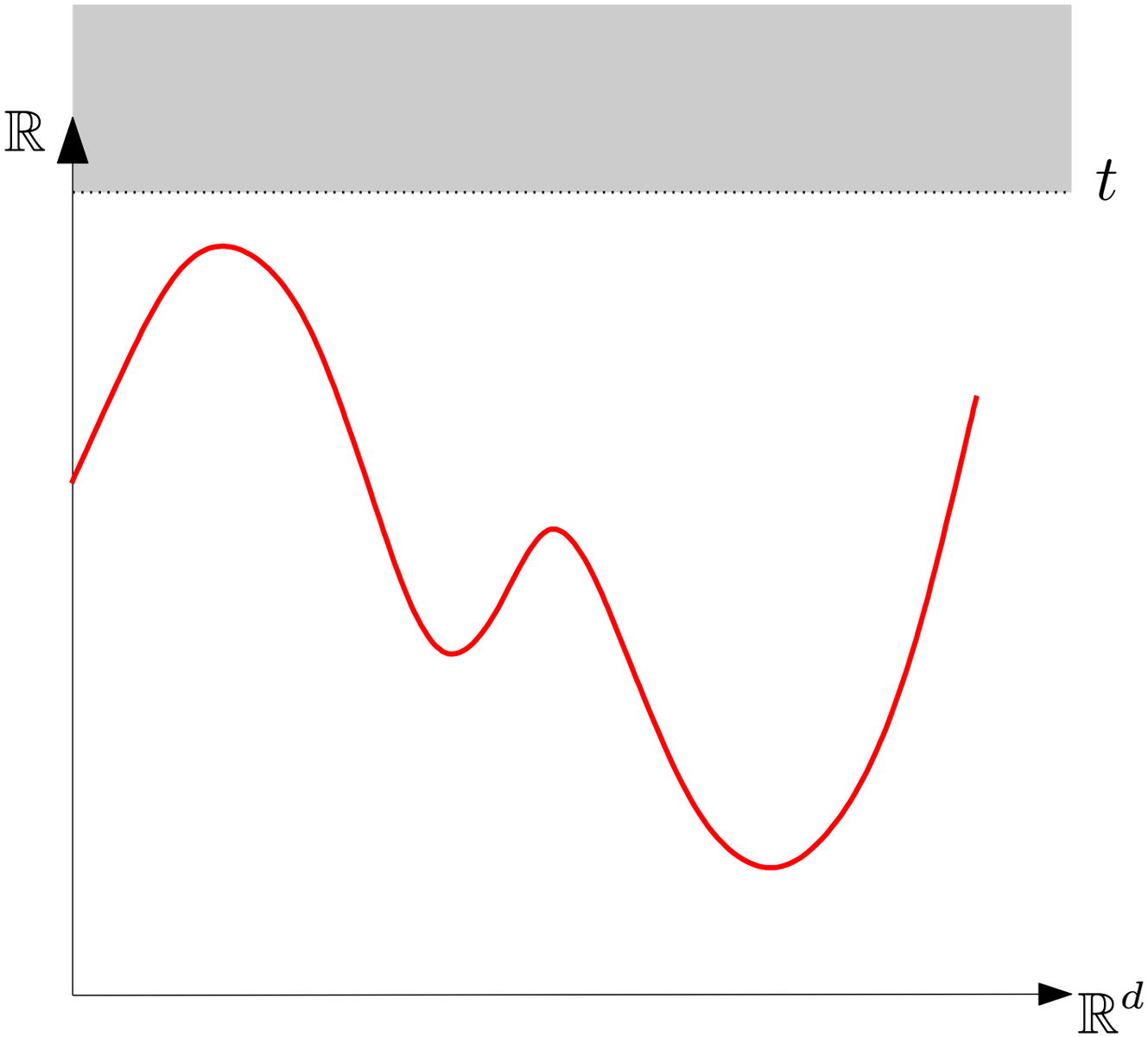


source: Wikipedia (https://commons.wikimedia.org/wiki/File:Relative-height.svg)

Persistence/prominence of peaks

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

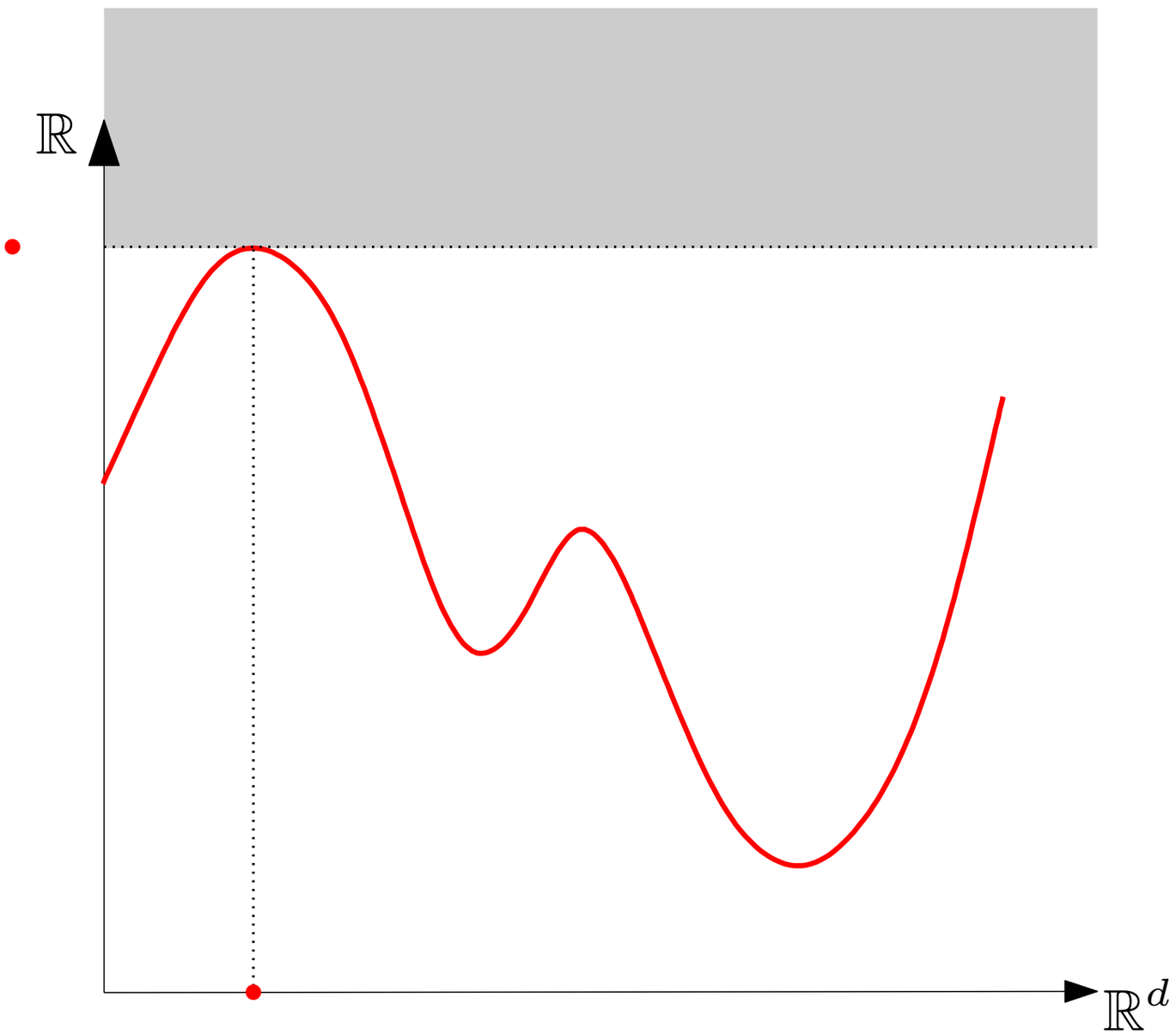
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

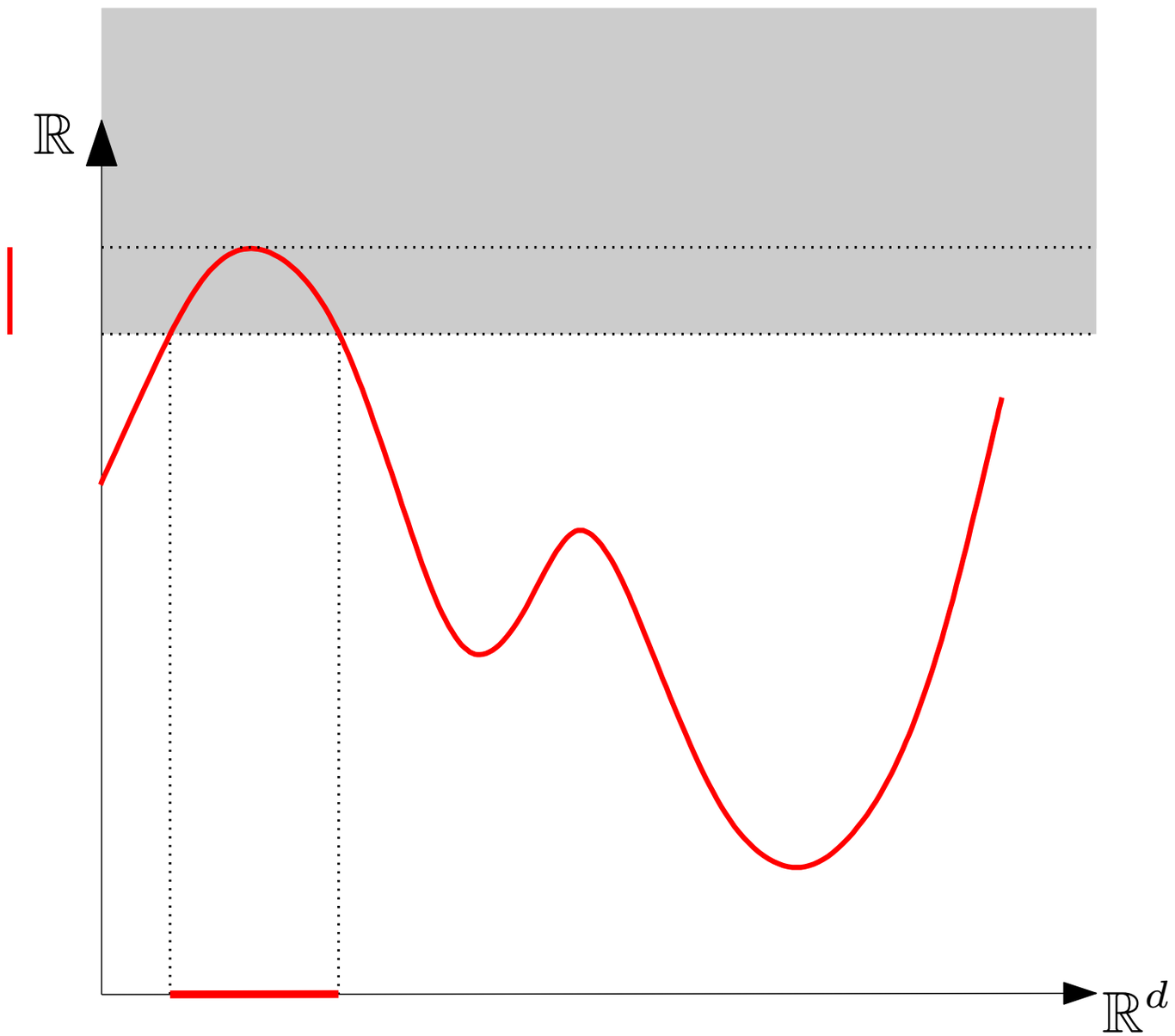
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

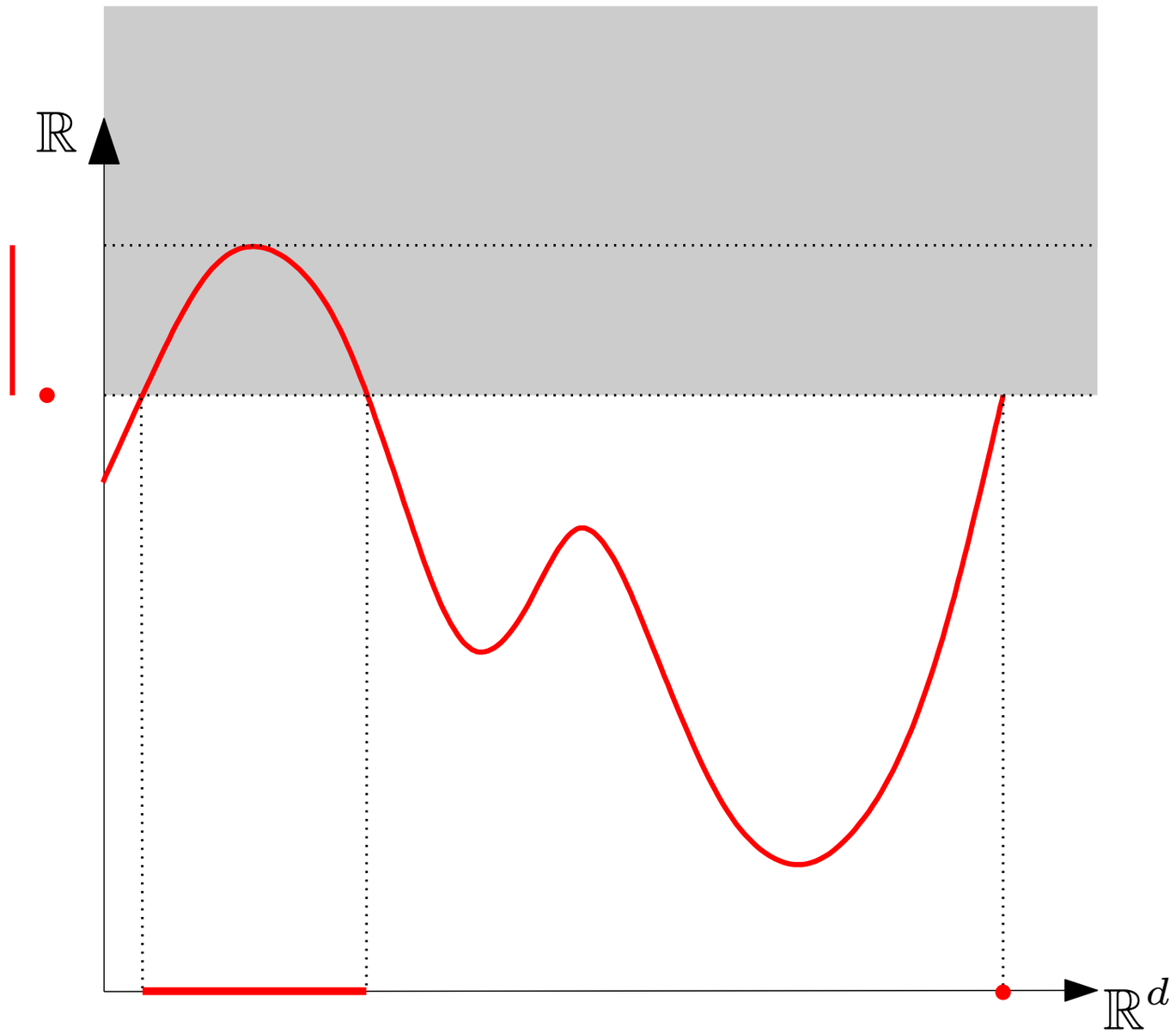
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

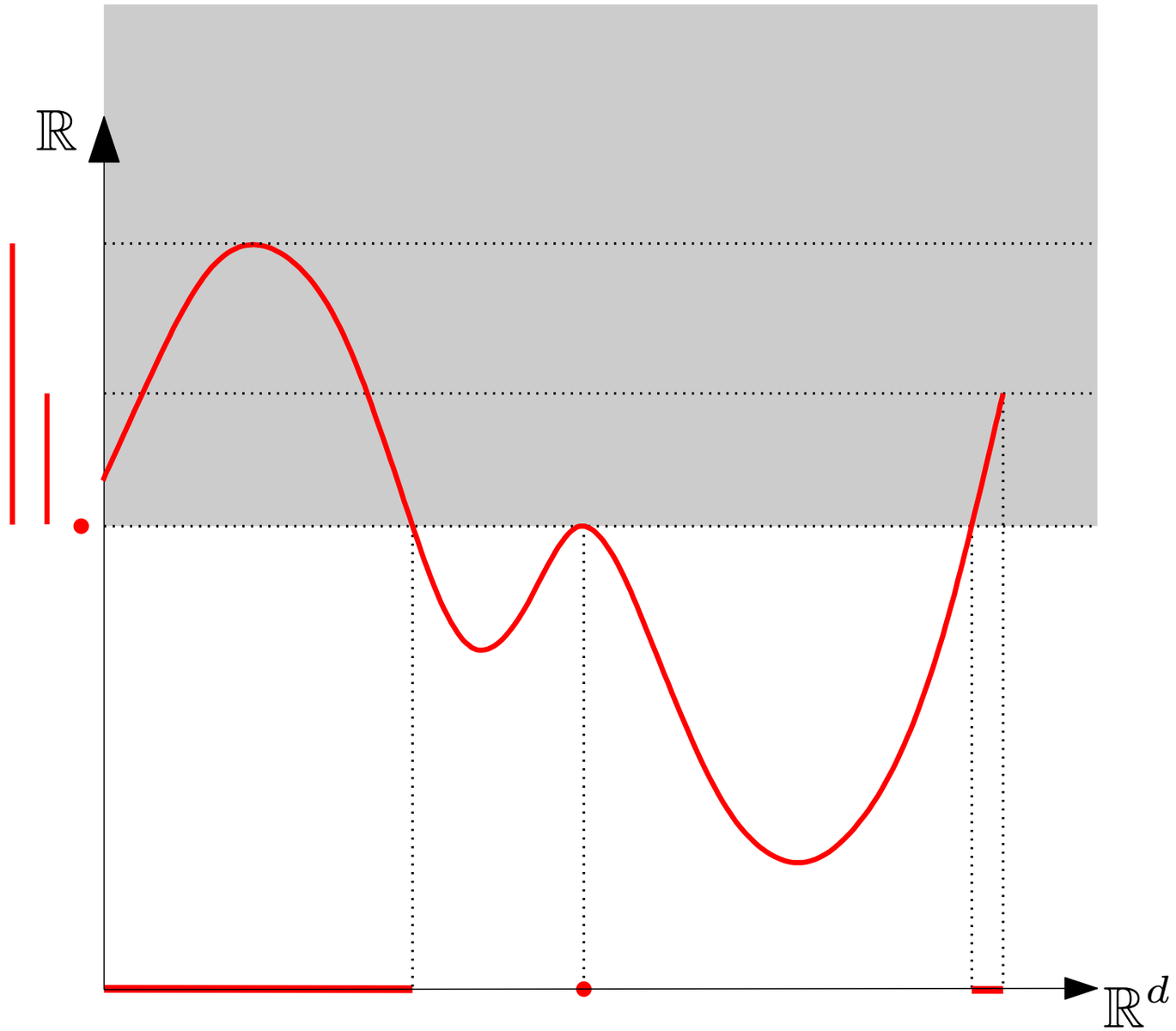
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

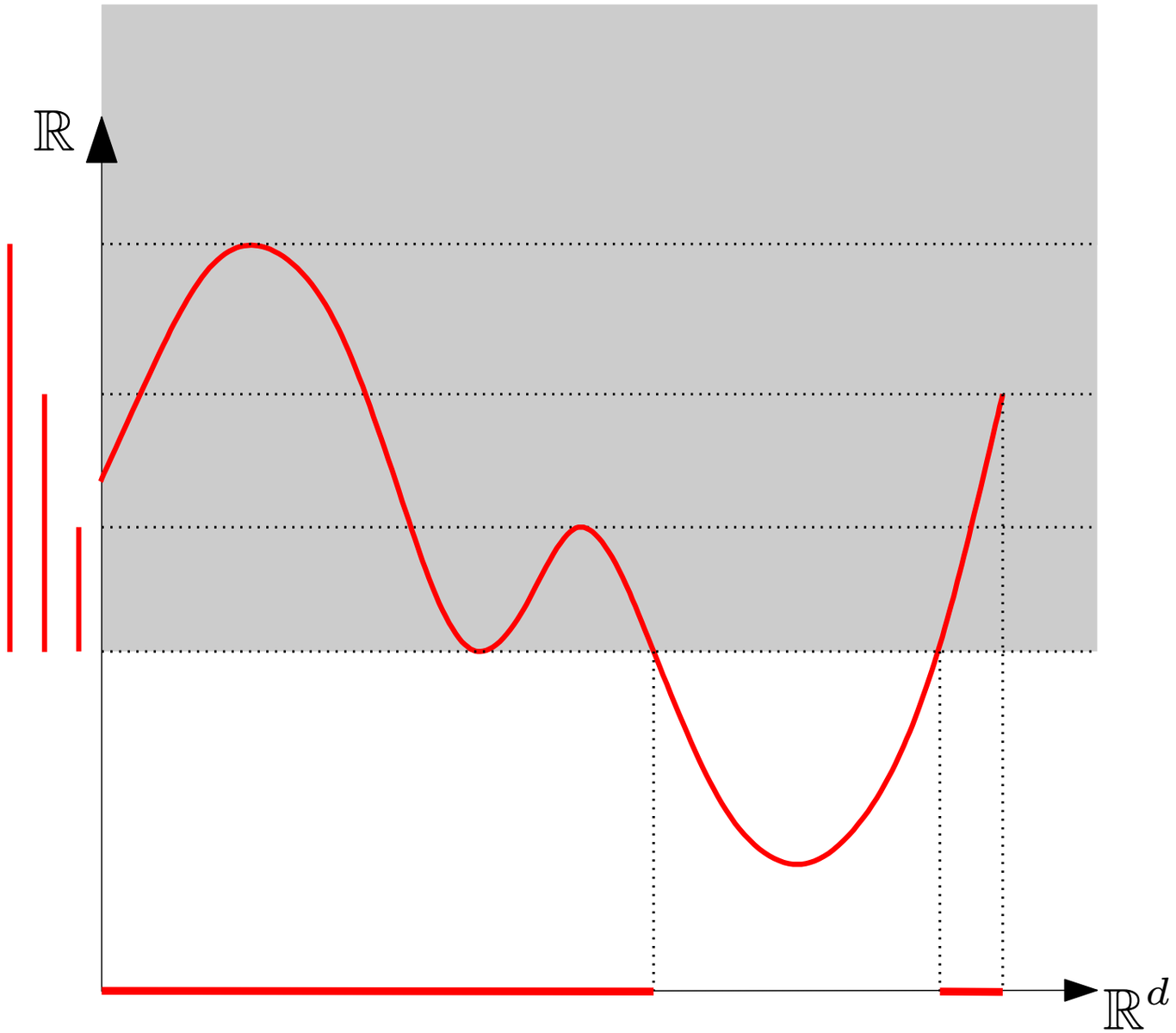
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

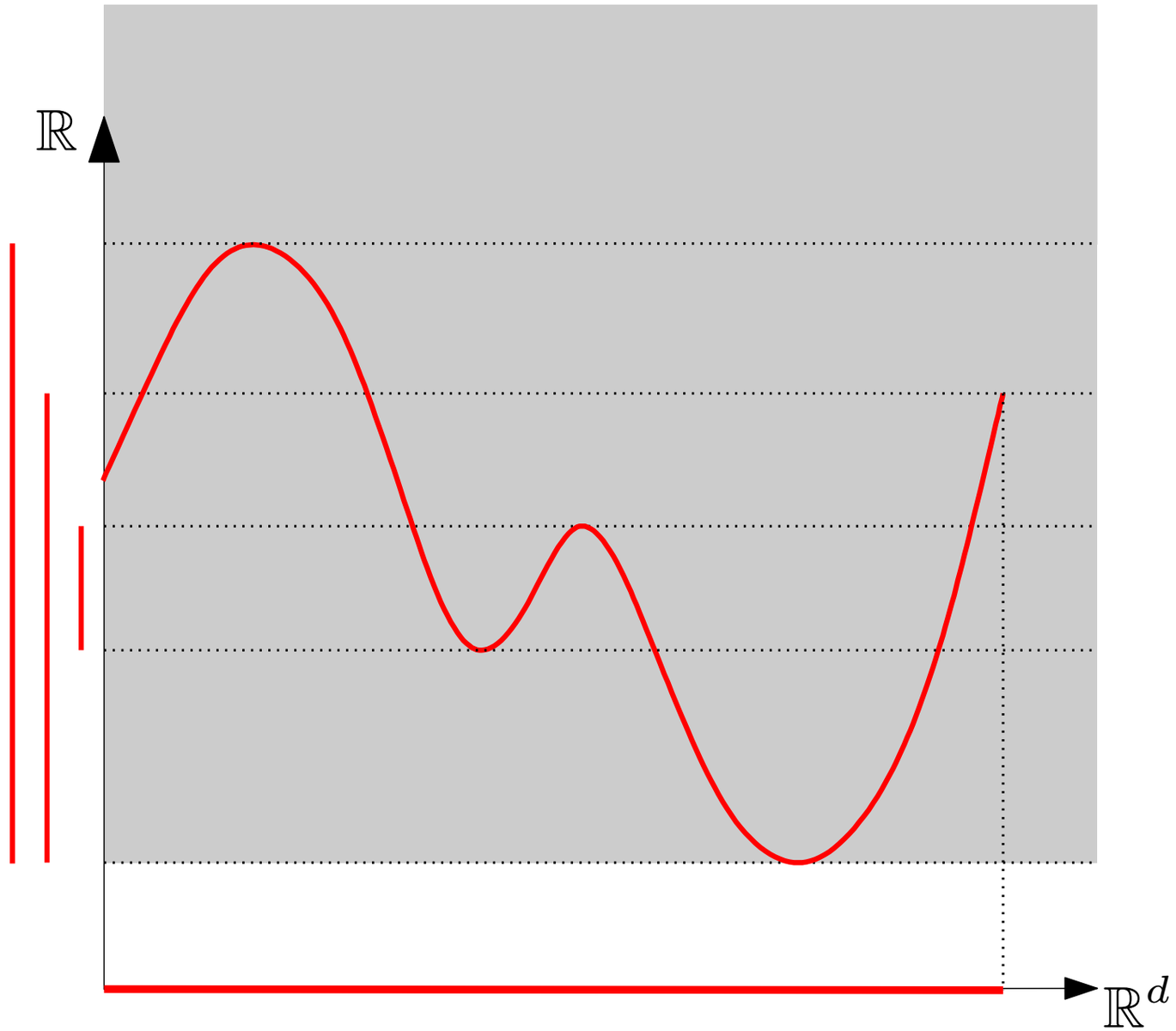
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

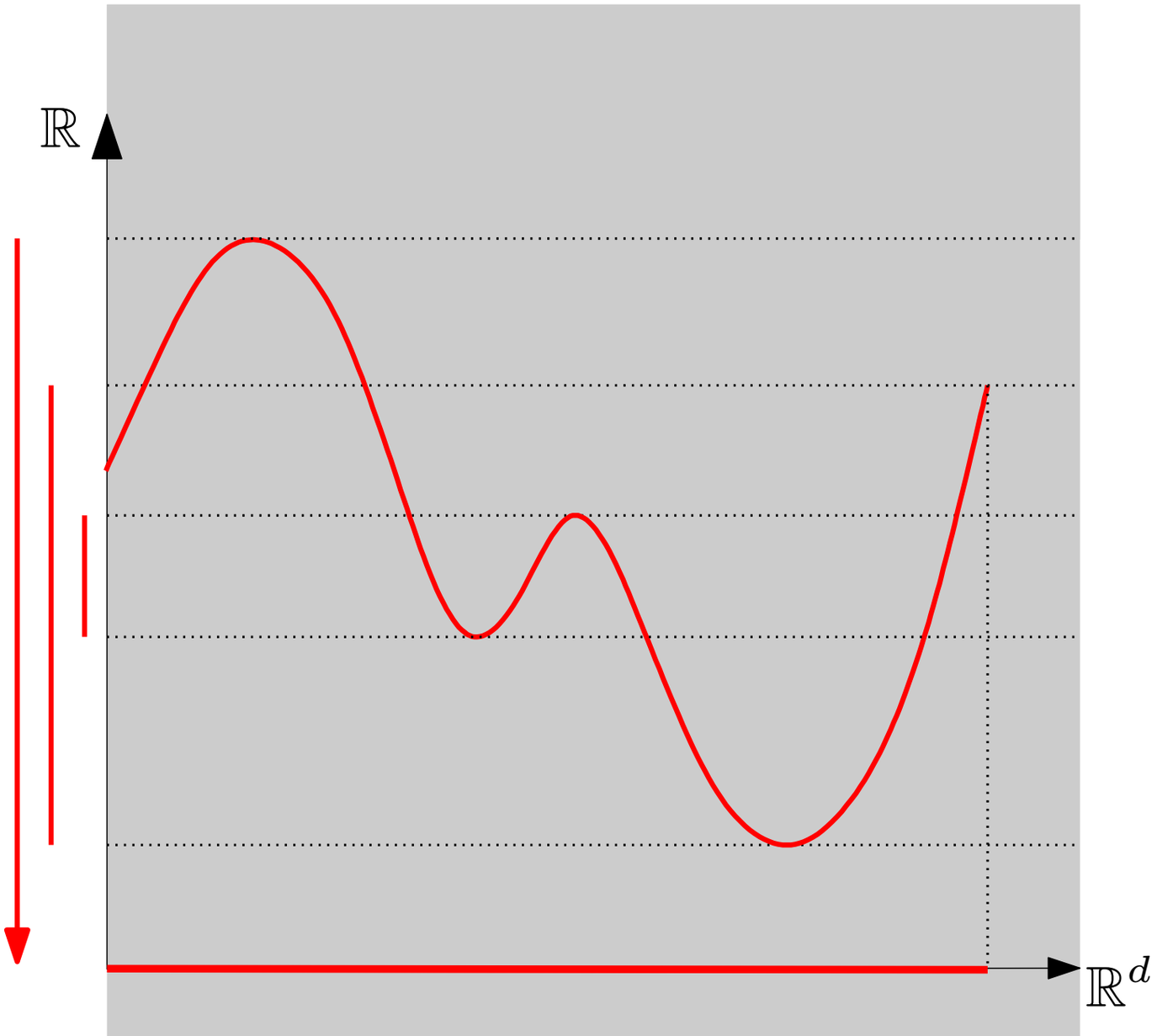
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family



Persistence/prominence of peaks

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

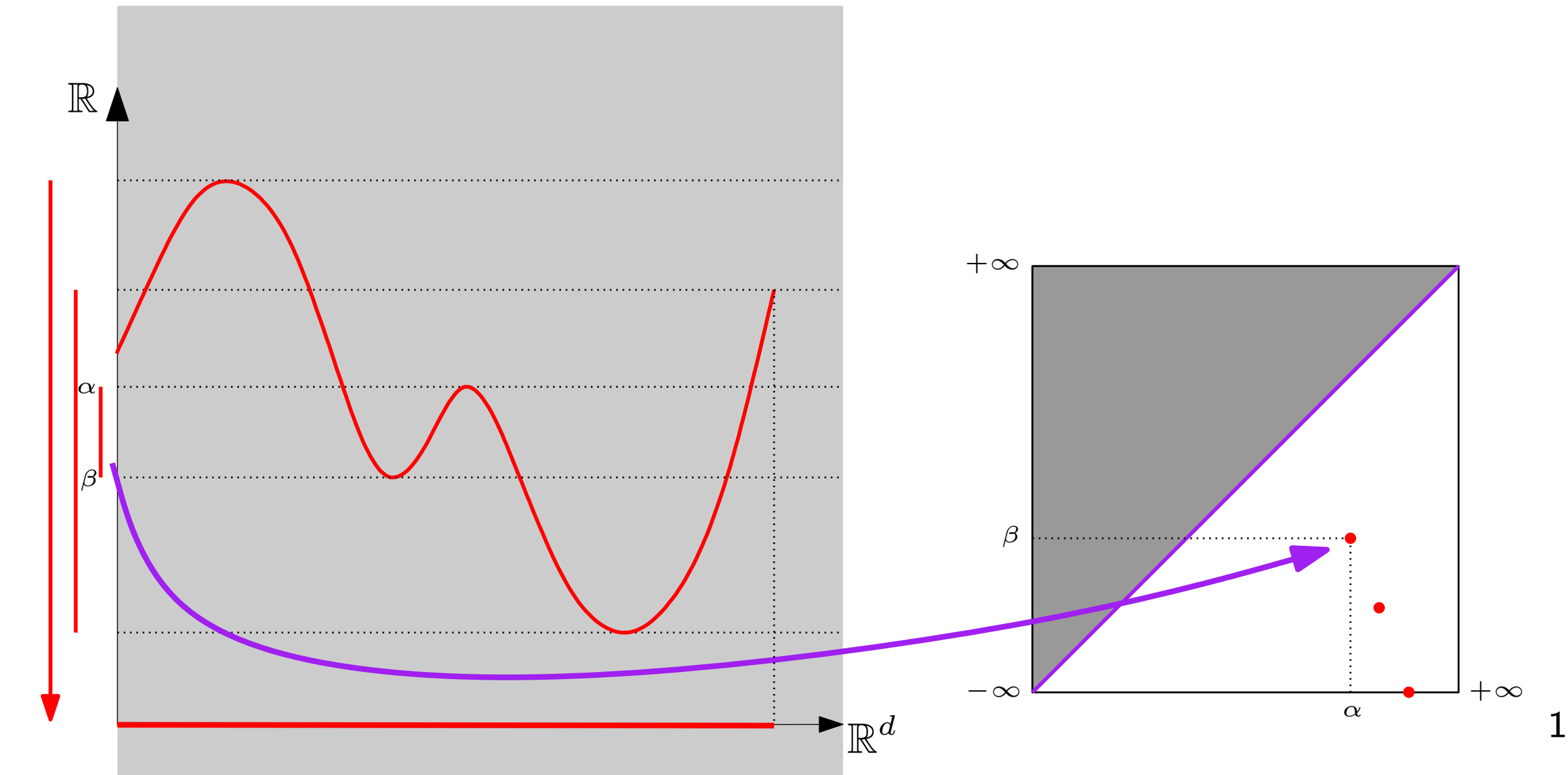
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Persistence/prominence of peaks

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

- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



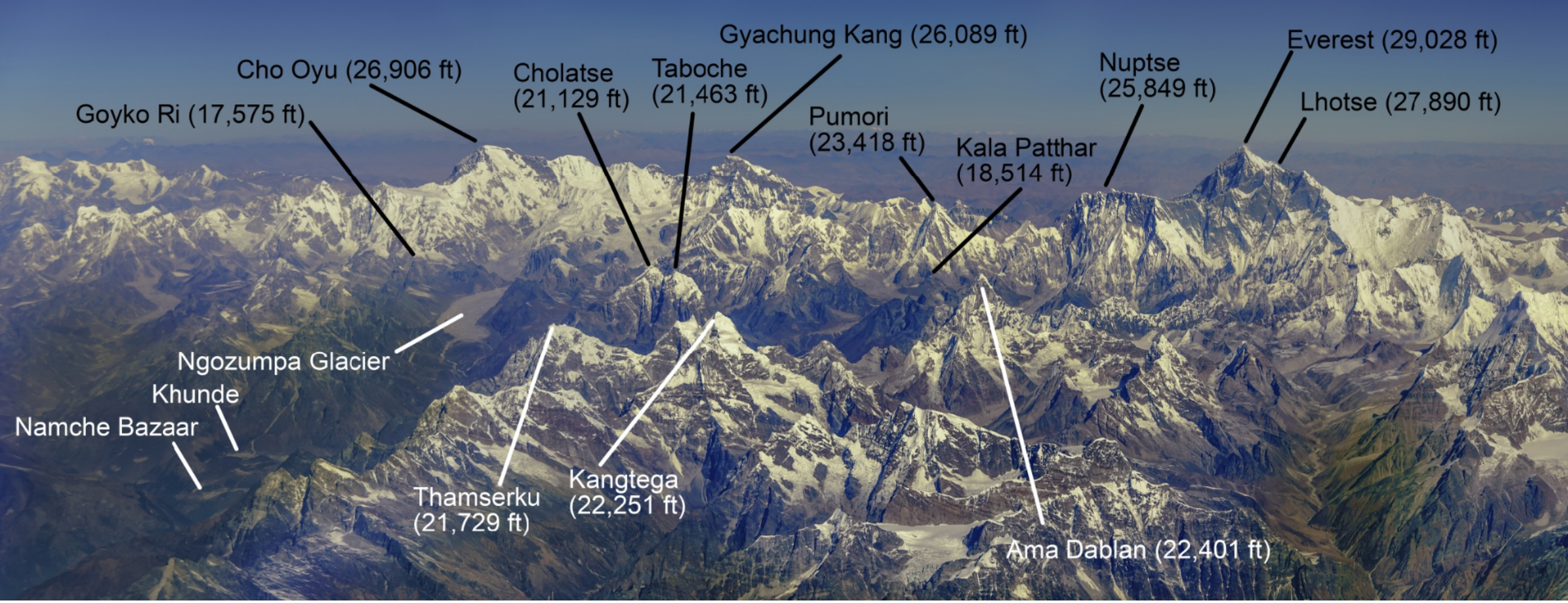


photo credits: Jim Block (http://www.gigapan.com/gigapans/198046/)

summit	continent	altitude (m)	**prominence** (m)	*parent*
Mount Everest	Asia	8849	8849	—
Aconcagua	Americas	6960	6960	—
Denali	Americas	6191	6155	Aconcagua
Mount Kilimanjaro	Africa	5895	5885	Mount Everest
Pico Cristóbal Colón	Americas	5700	5509	Aconcagua
Mount Logan	Americas	5959	5250	Denali
Pico de Orizaba	Americas	5636	4922	Mount Logan
Vinson Massif	Antarctica	4892	4892	—
Puncak Jaya	Oceania	4884	4884	—
Mount Elbrouz	Europe	5642	4741	Mount Everest

animation: https://upload.wikimedia.org/wikipedia/commons/3/35/World_top_peak_prominence.svg

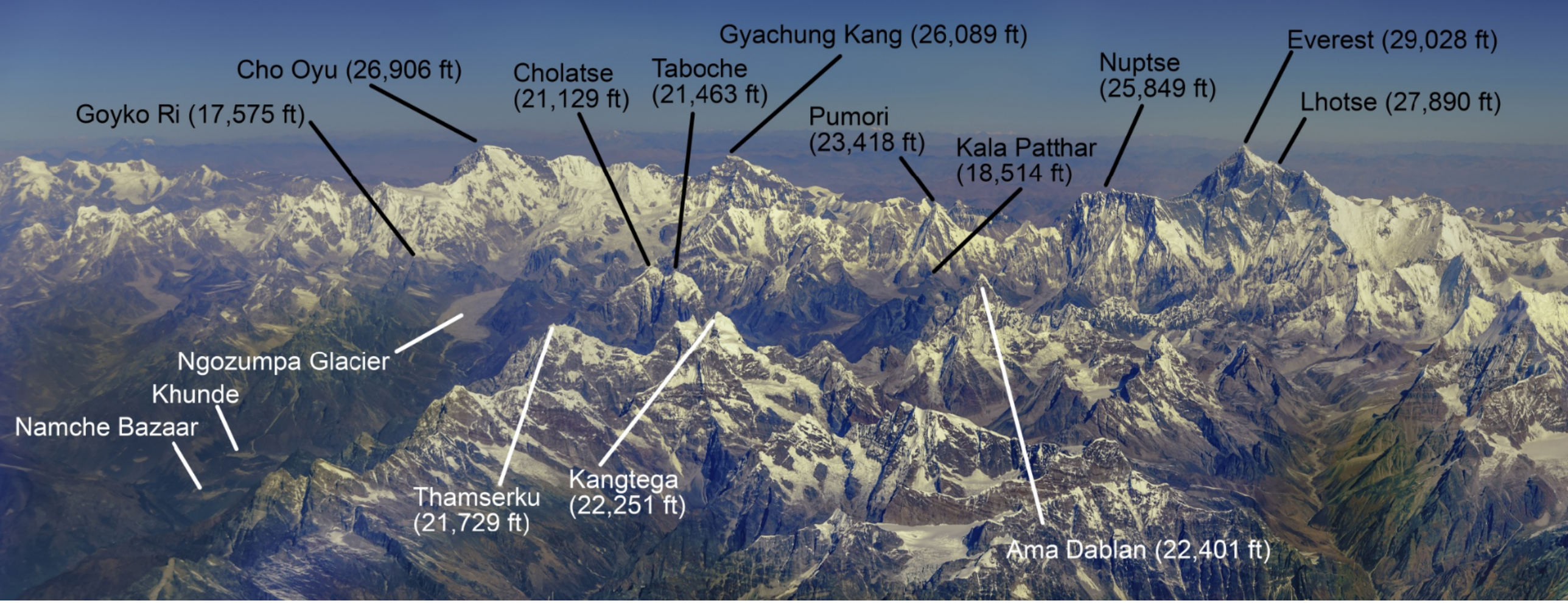
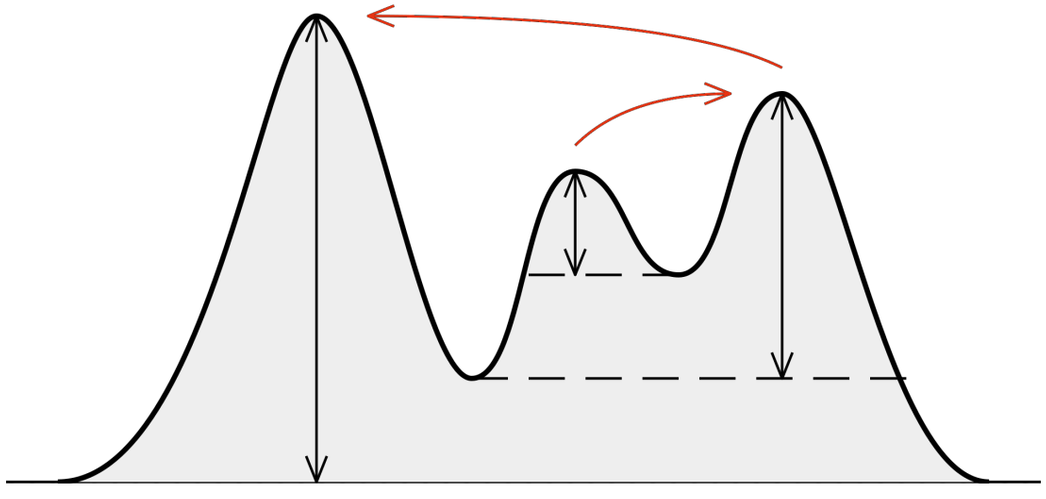


photo credits: Jim Block (http://www.gigapan.com/gigapans/198046/)

parent of a peak
(according to alpinists):

highest peak accessible through the highest pass

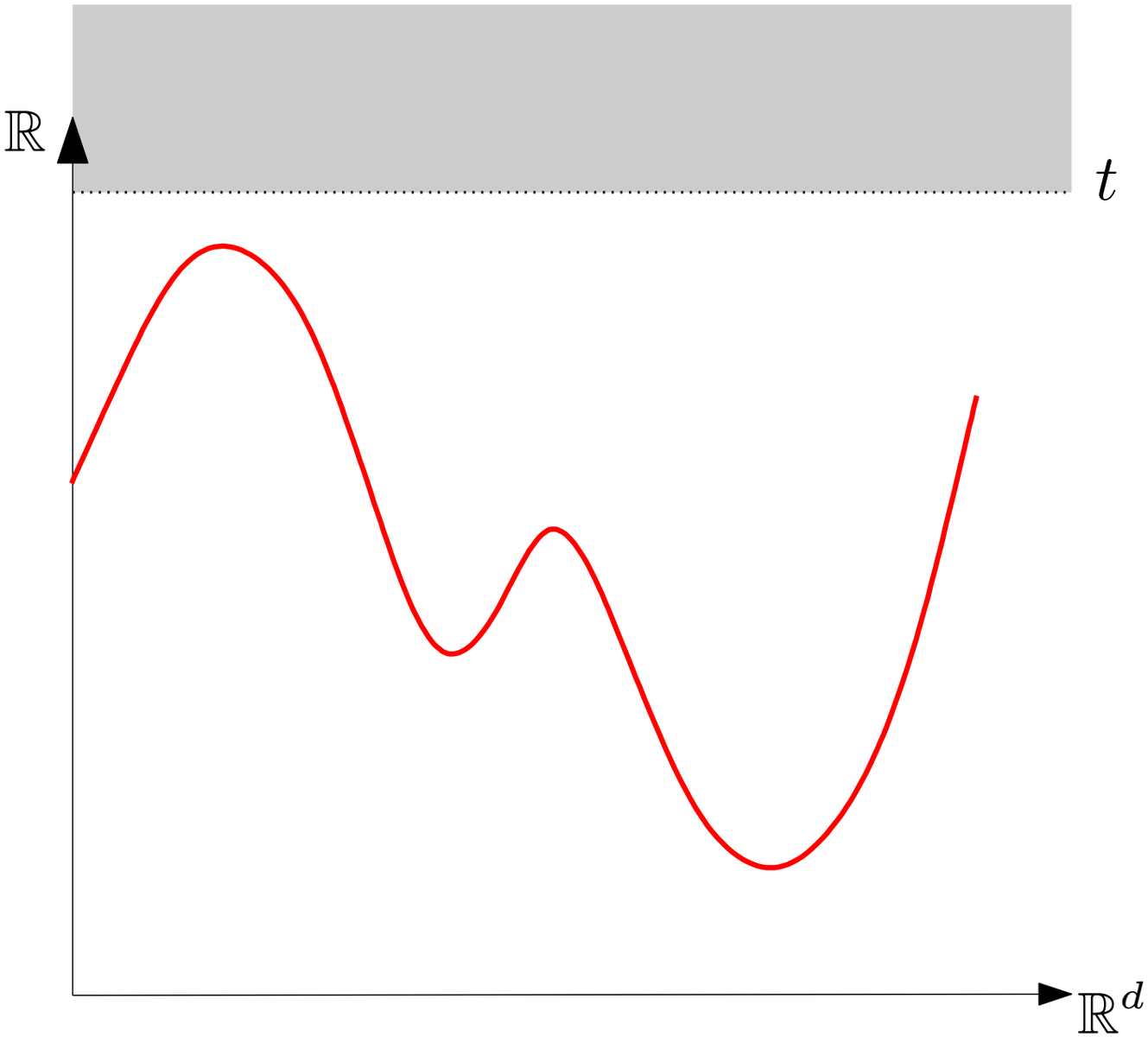


source: Wikipedia (https://en.wikipedia.org/wiki/Topographic_prominence)

Parents of peaks

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

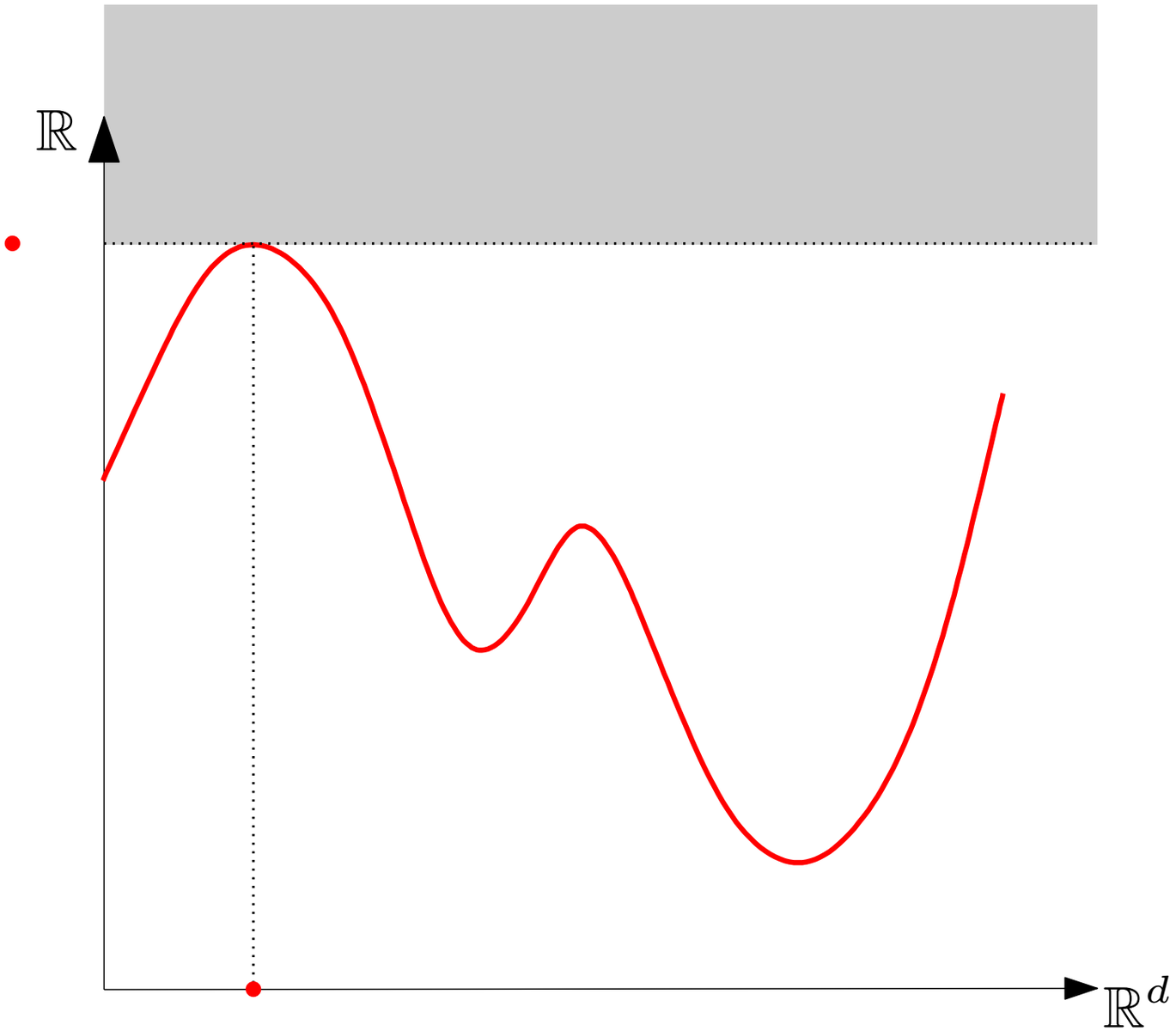
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

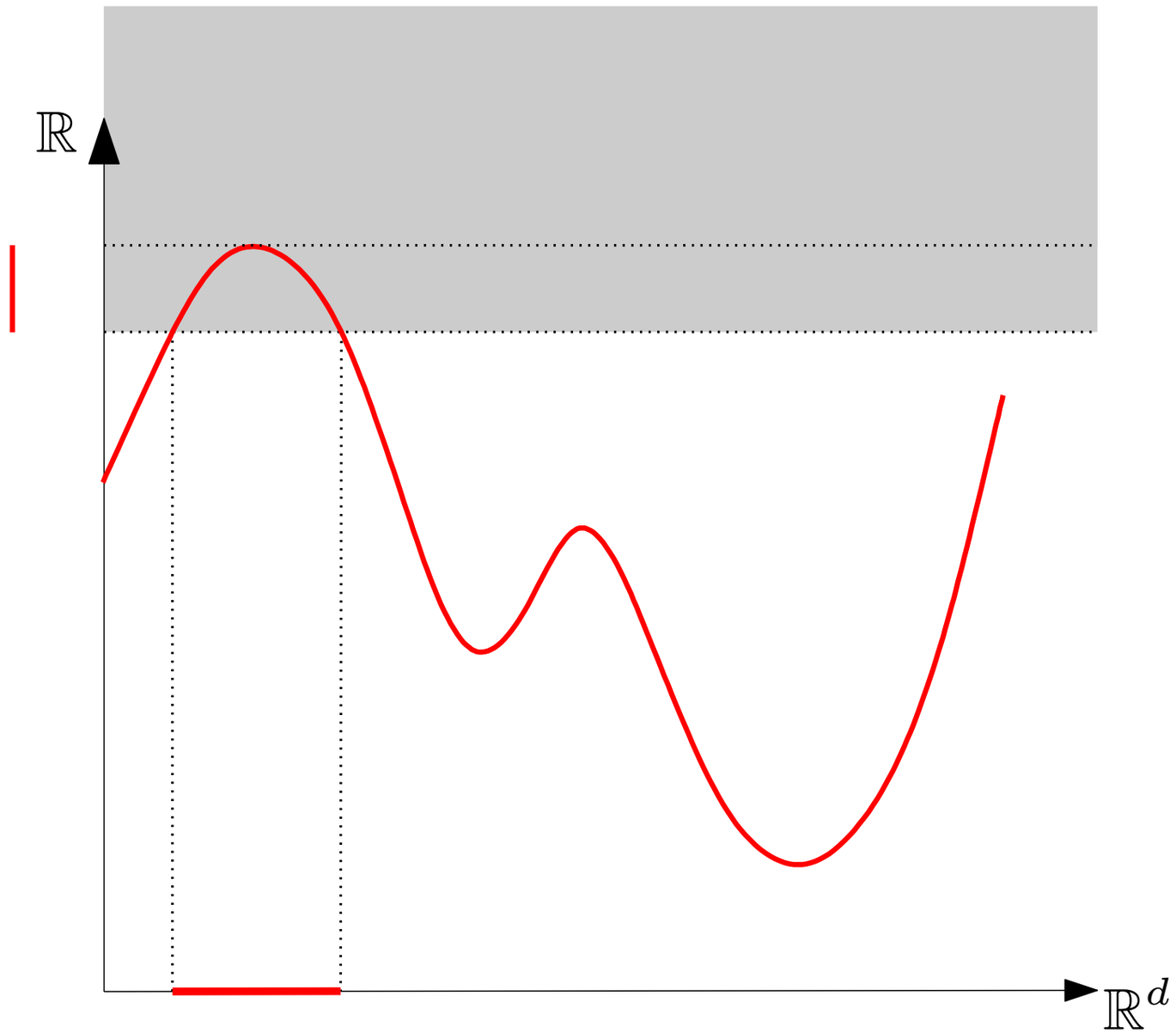
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

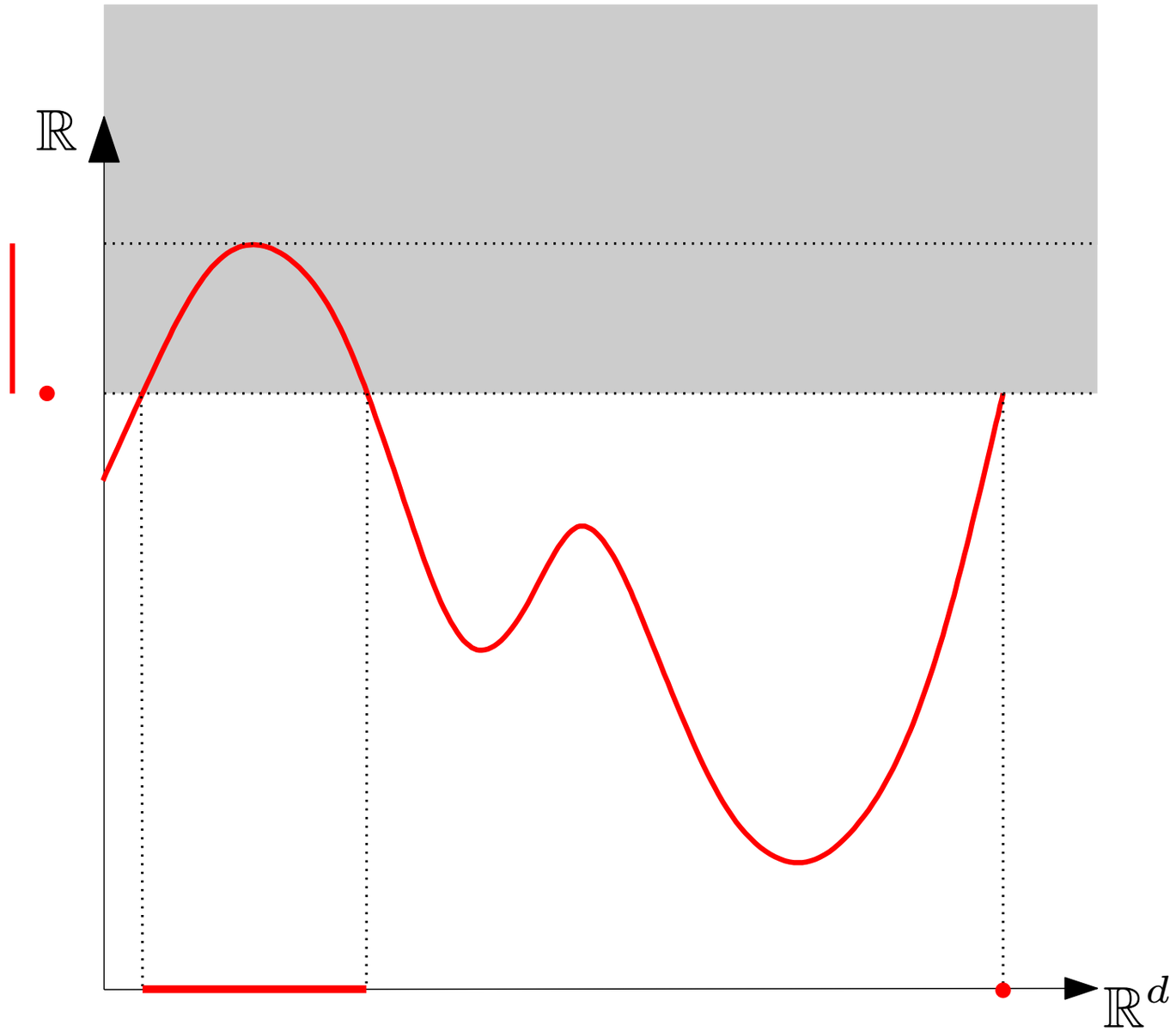
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

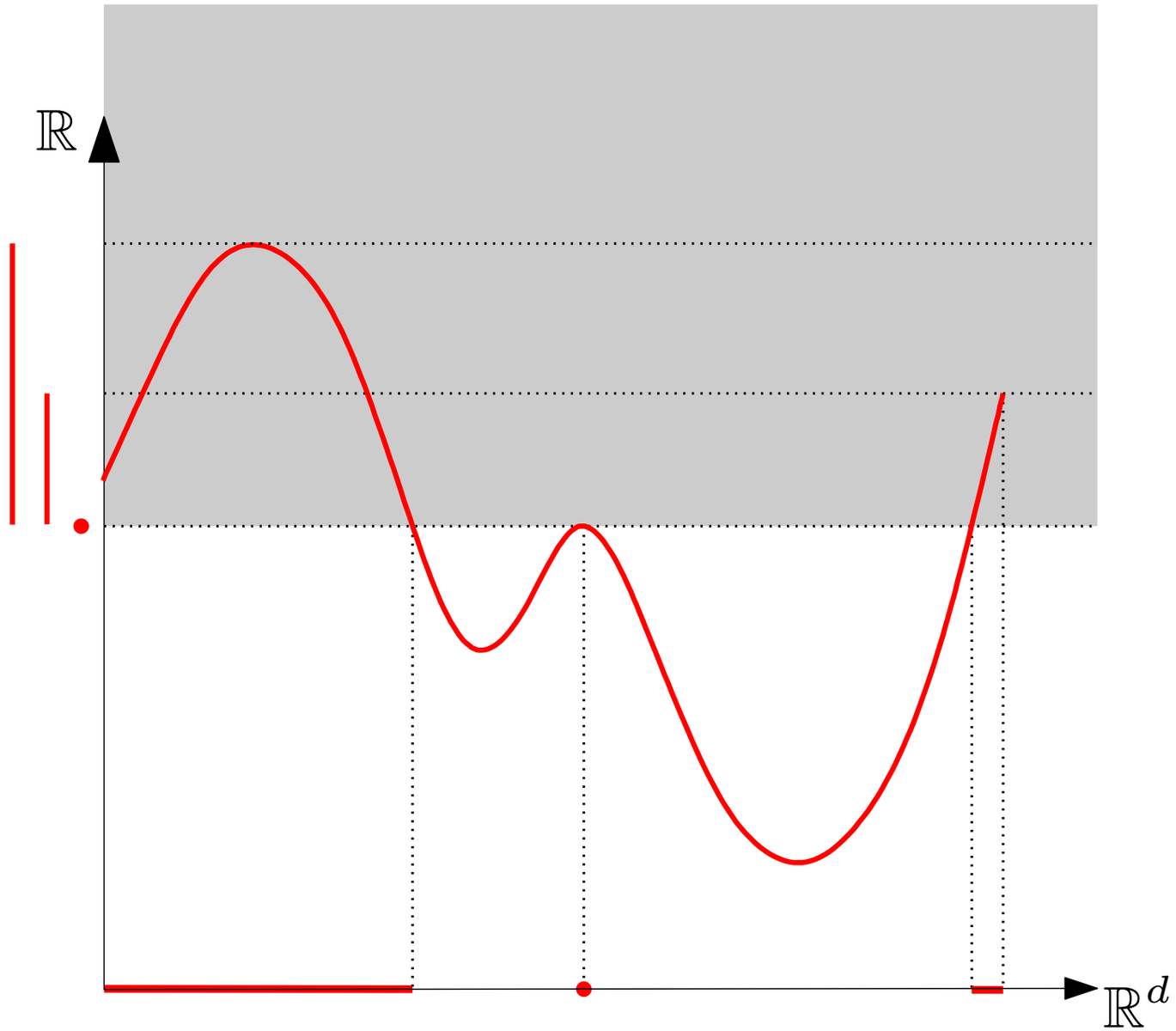
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

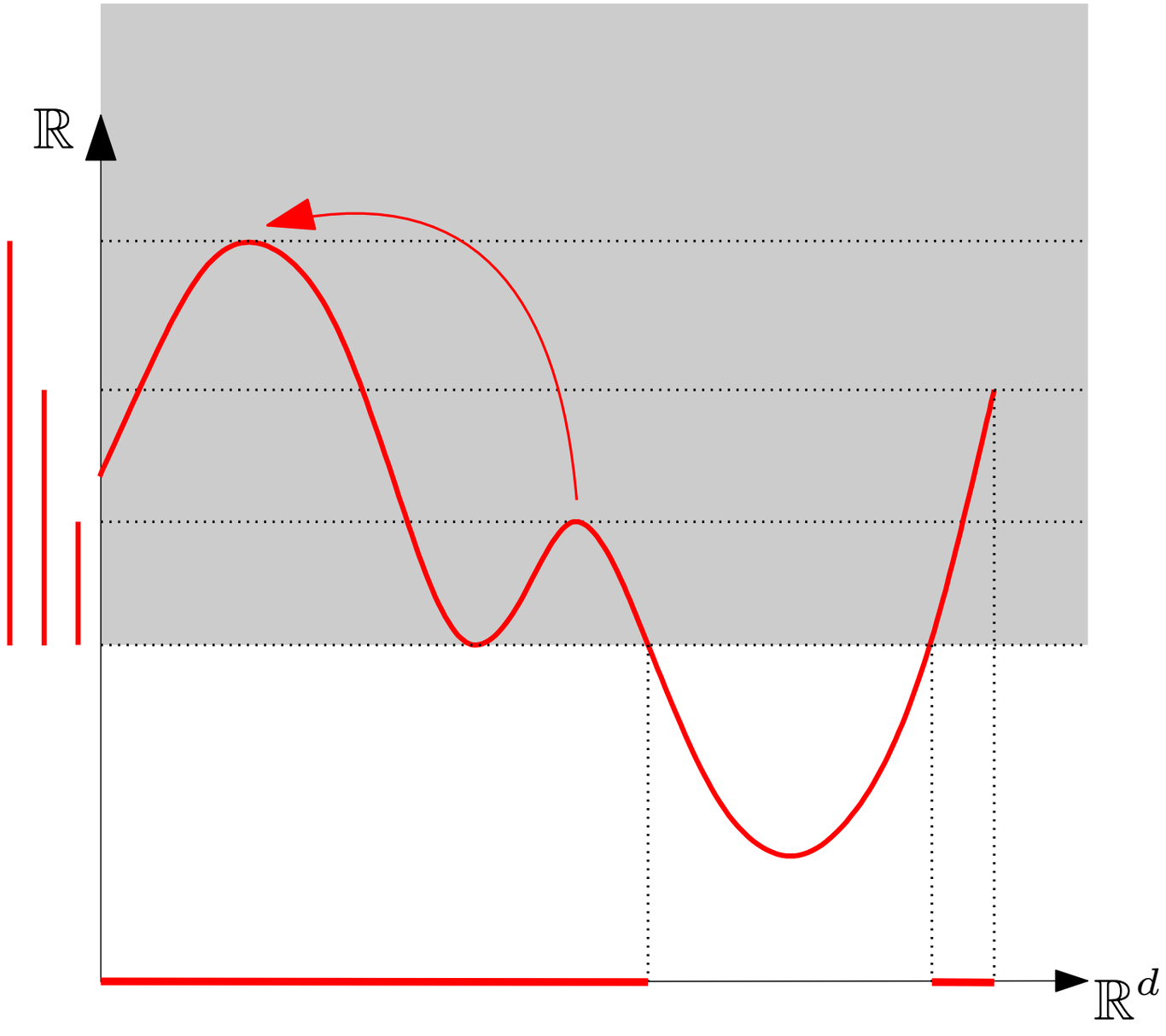
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

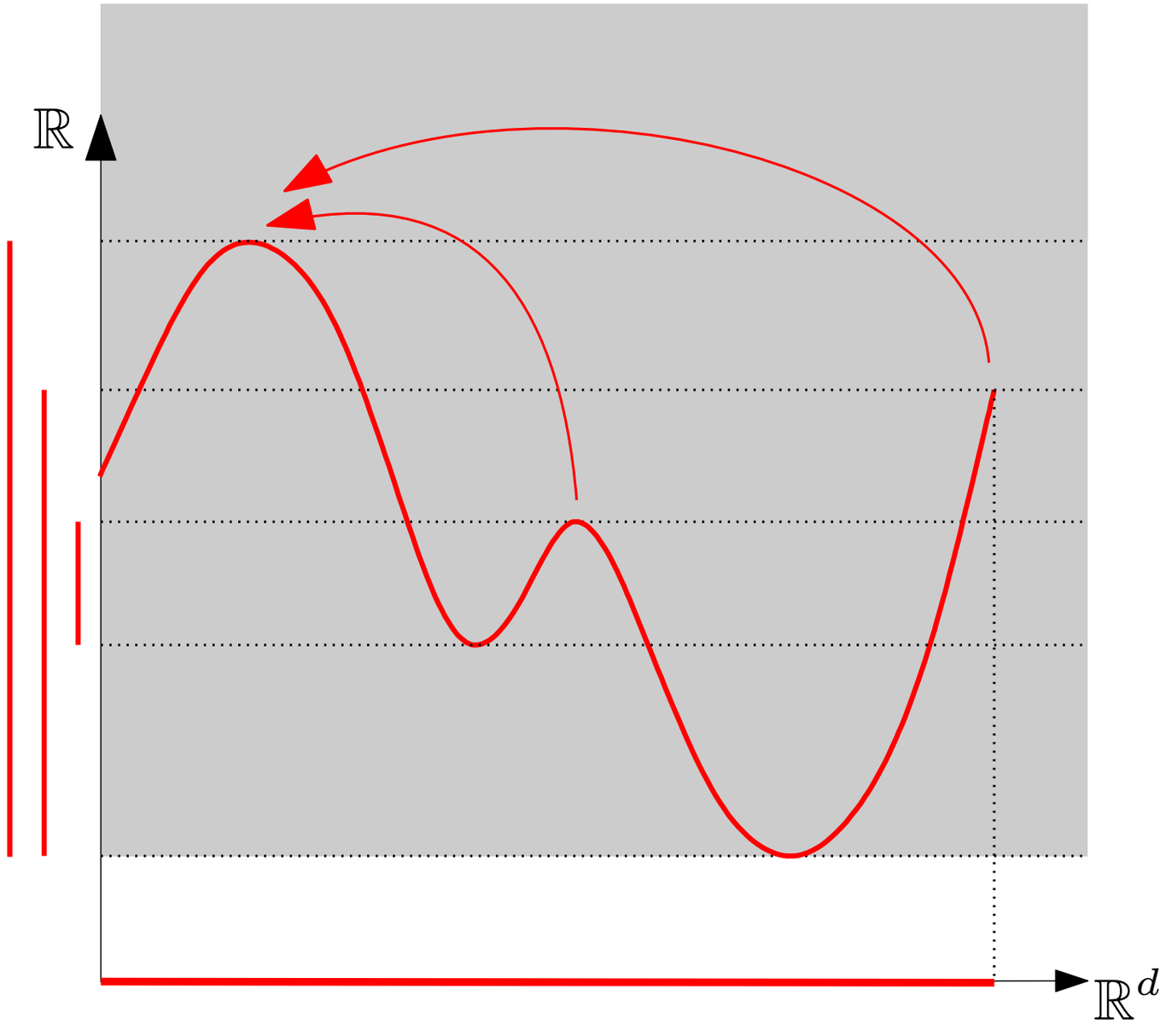
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

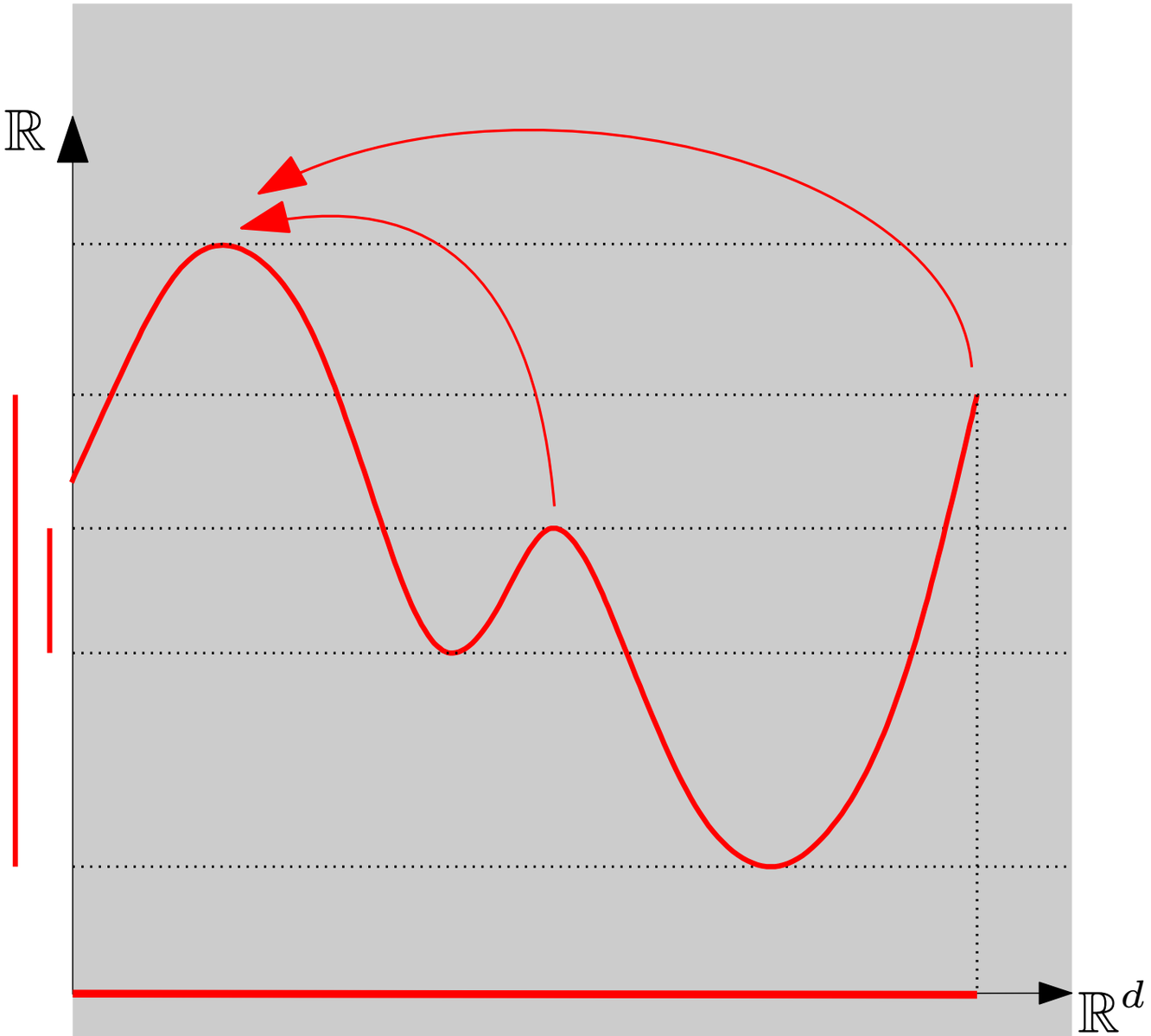
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

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

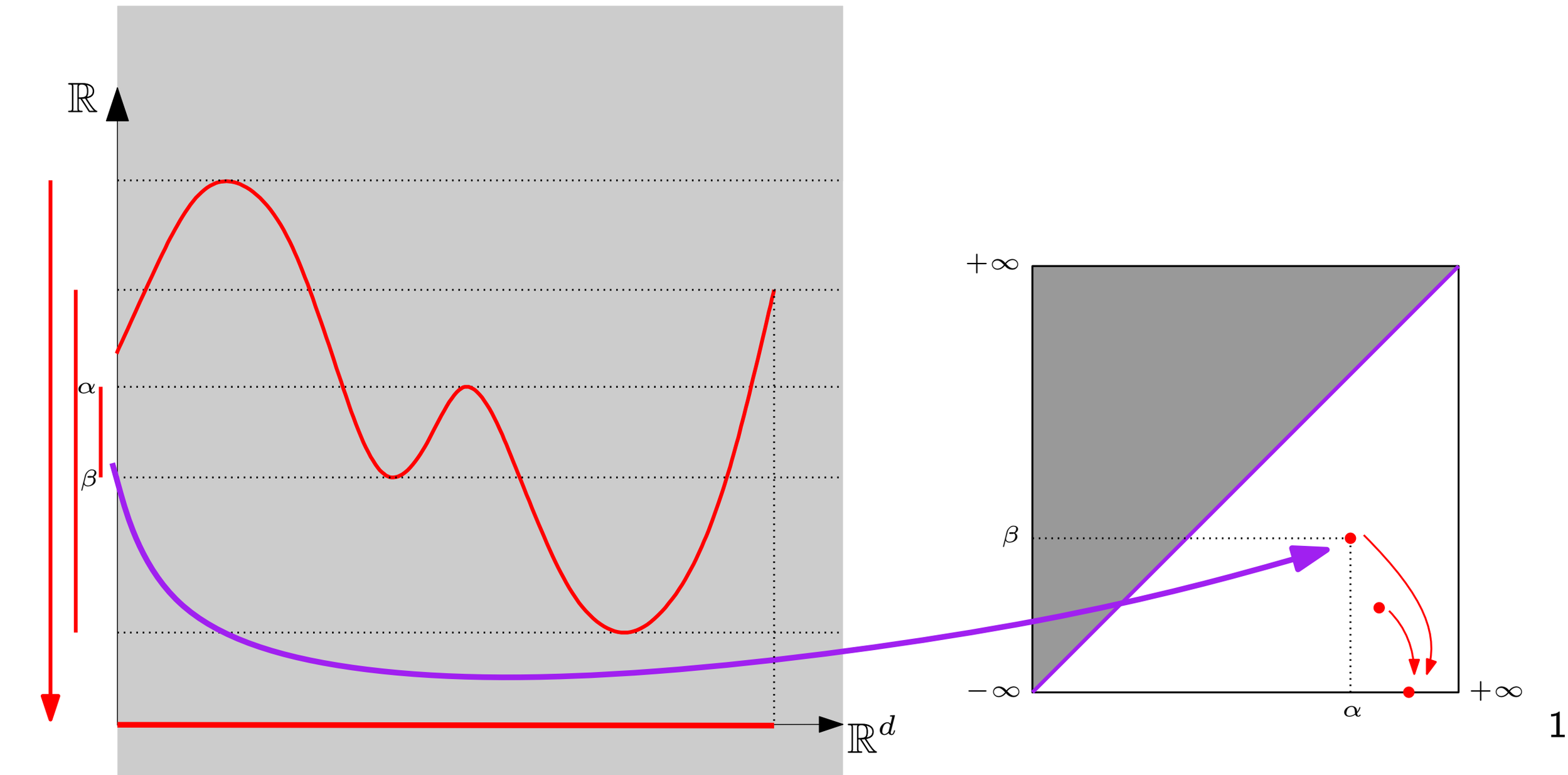
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Parents of peaks

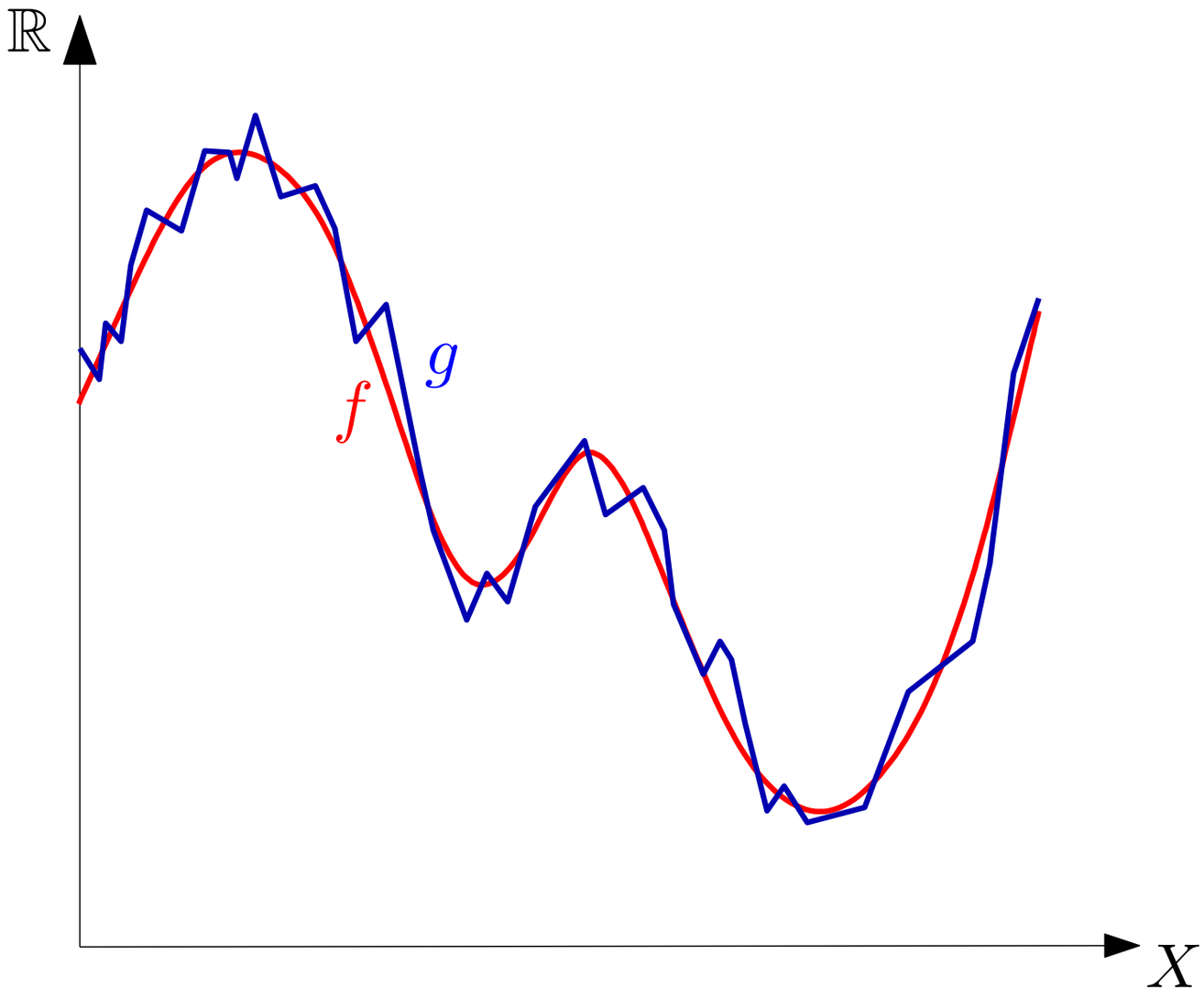
Given $f : X \to \mathbb{R}$:

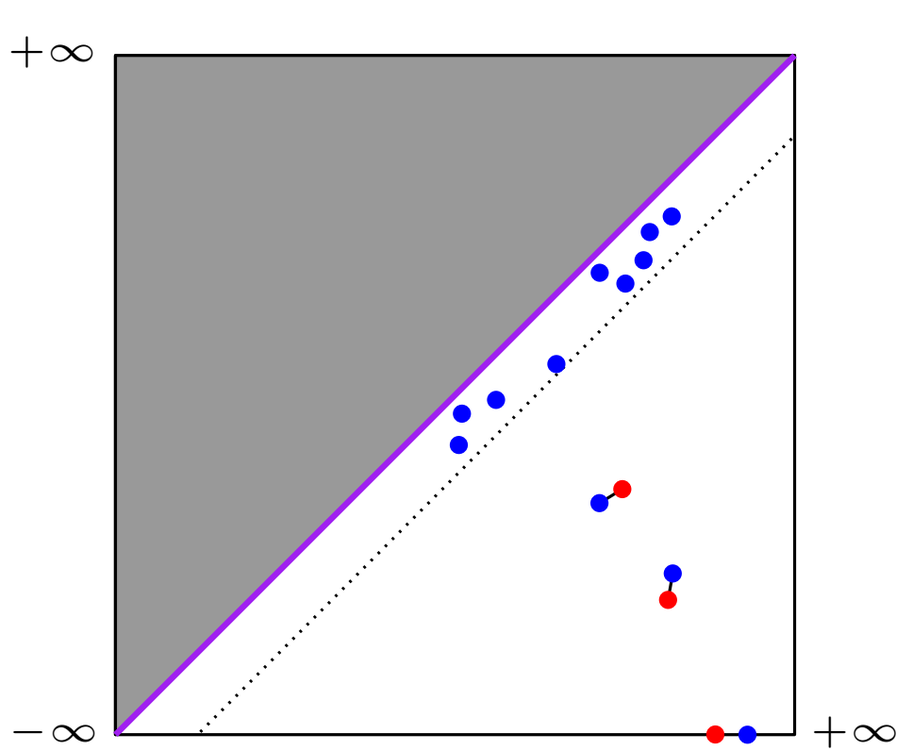
- Nested family (filtration) of **superlevel-sets** $f^{-1}([t, +\infty))$ for t from $+\infty$ to $-\infty$
- Track evolution of connected components throughout the family
- Multiset of intervals (barcode) encodes appearance/merger of connected components



Persistence diagrams and stability

Theorem (Stability): [Cohen-Steiner, Edelsbrunner, Harer 2005]
For any $f, g\colon X \to \mathbb{R}$ *tame*, $\mathrm{d}_b(\mathrm{Dg}\,(f), \mathrm{Dg}\,(g)) \leq \sup_{x\in X} |f(x) - g(x)|$.

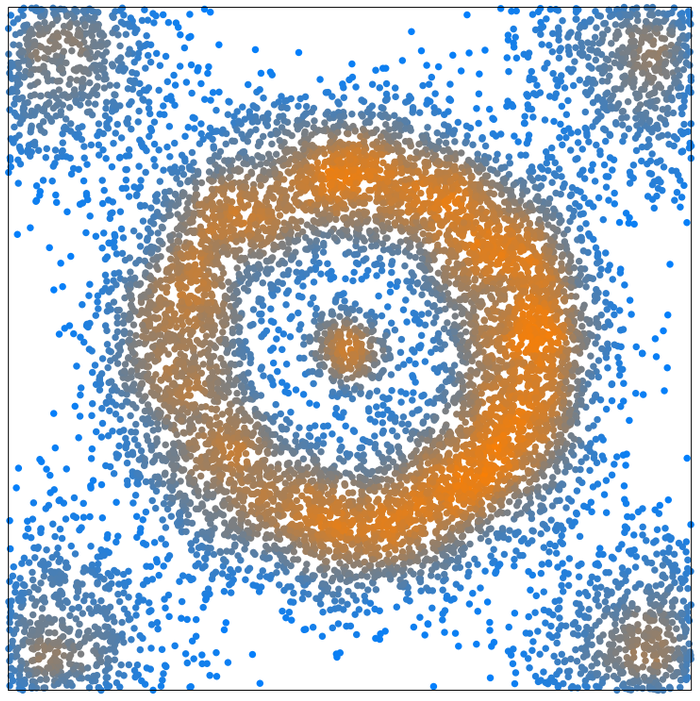




In clustering: approximate $\mathrm{Dg}\, f$ by $\mathrm{Dg}\, \hat{f}$

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

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



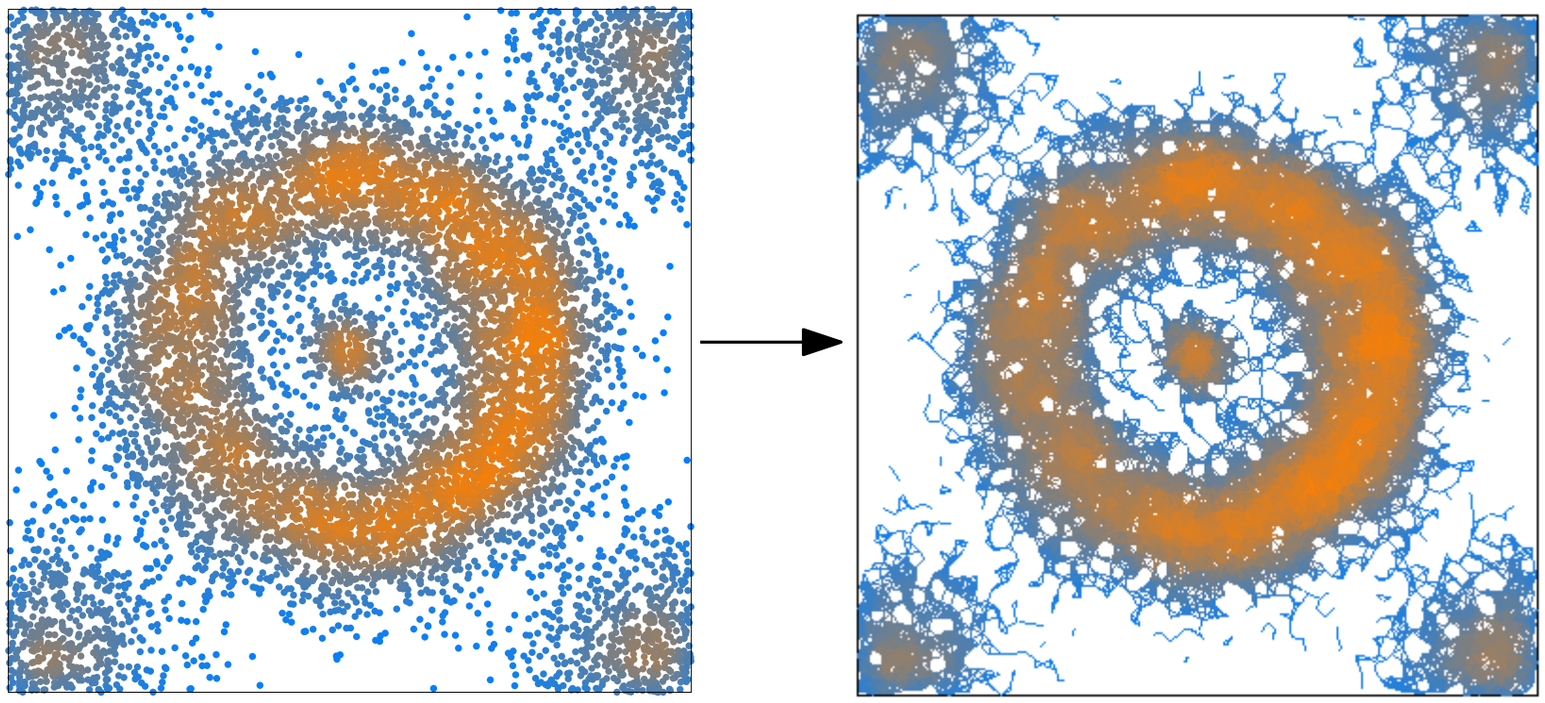
In clustering: approximate $\mathrm{Dg}\, f$ by $\mathrm{Dg}\, \hat{f}$

- 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*

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



In clustering: approximate $\mathrm{Dg}\, f$ by $\mathrm{Dg}\, \hat{f}$

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

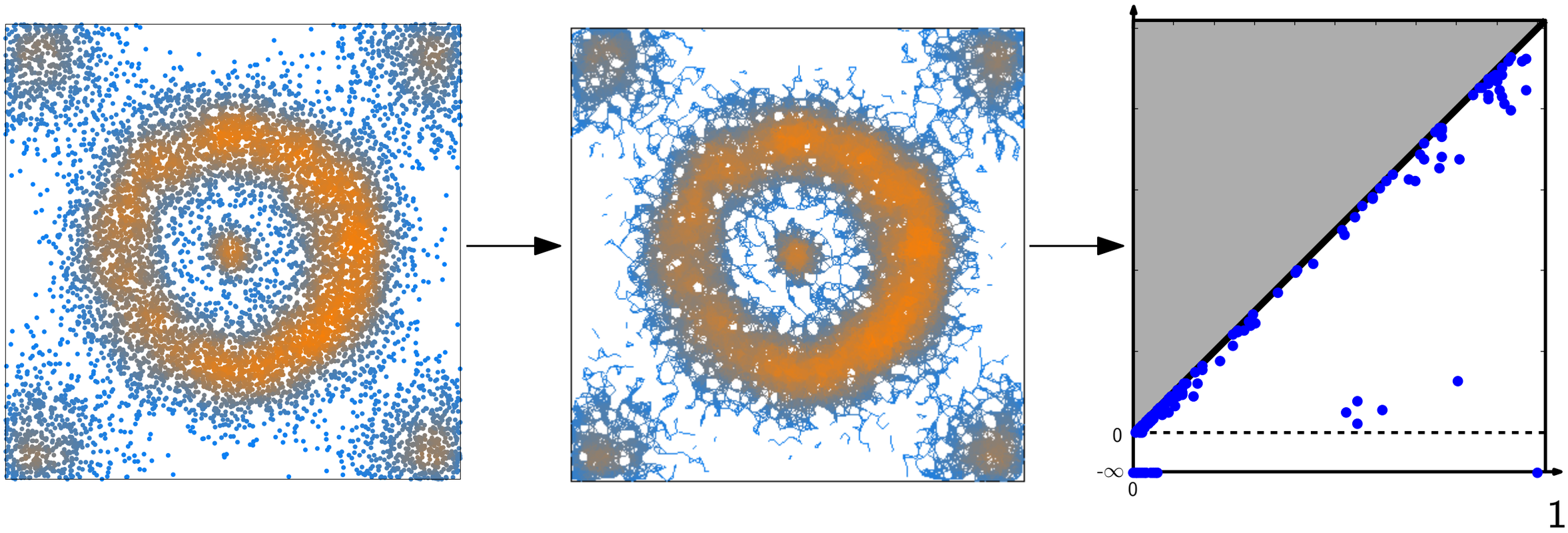
  (sort data points by **decreasing** estimated density values)

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

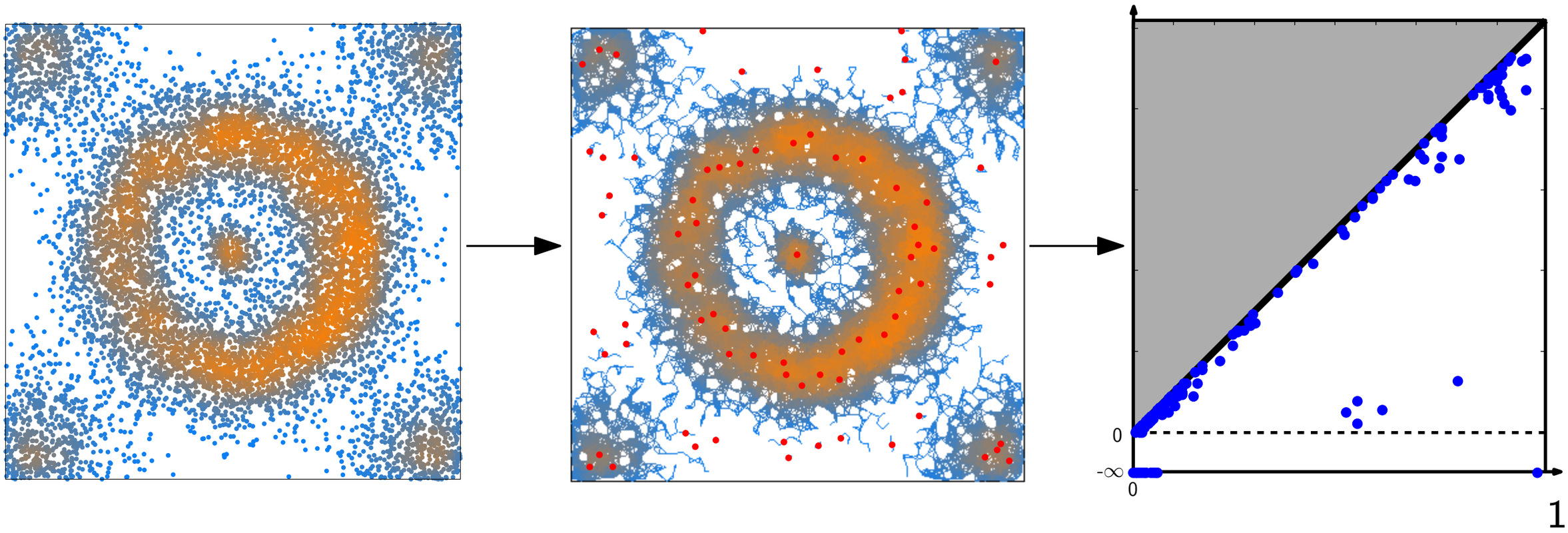
  $(\hat{f}([u,v]) = \min\{\hat{f}(u),\ \hat{f}(v)\})$

- Compute the 0-dimensional persistence diagram of this filtration

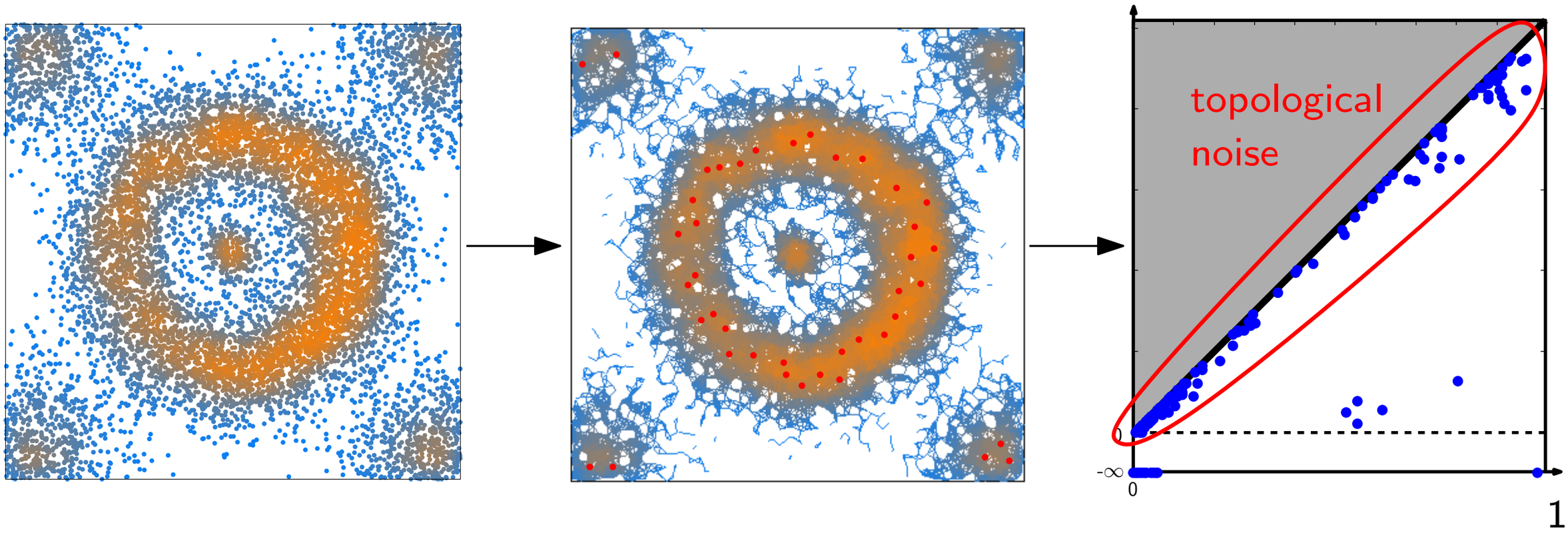
  (apply 0-dimensional persistence algorithm $\rightarrow$ union-find data structure)



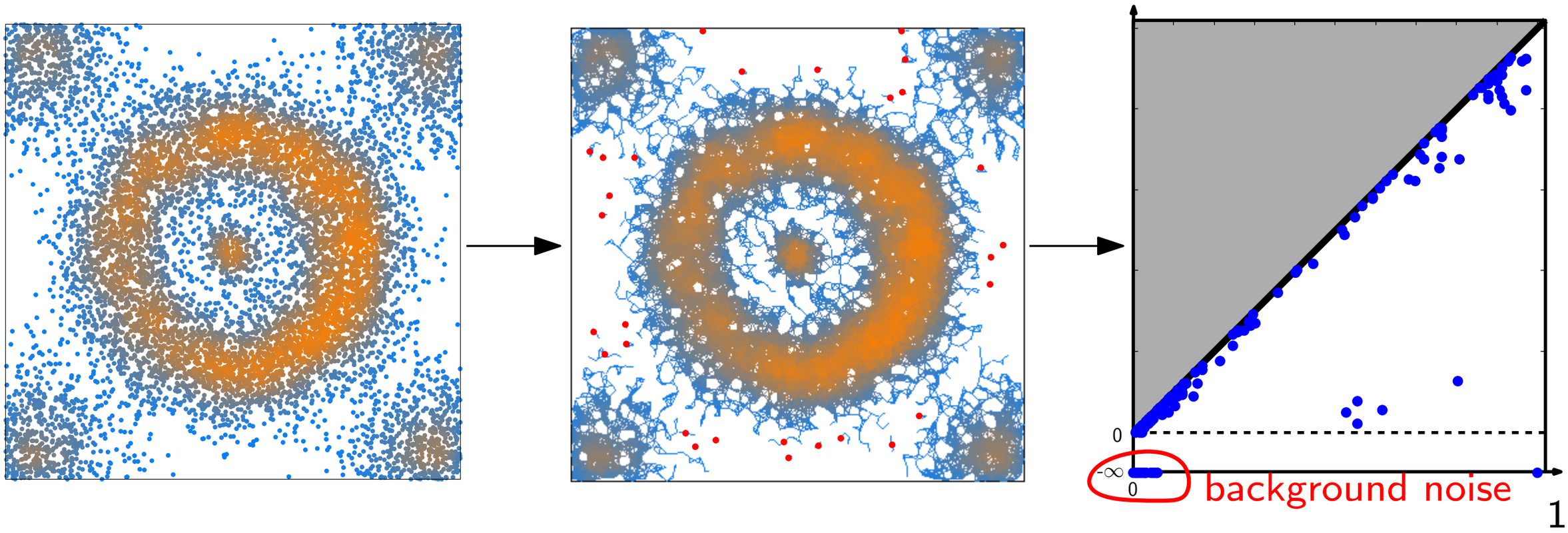
Estimating the Correct Number of Clusters



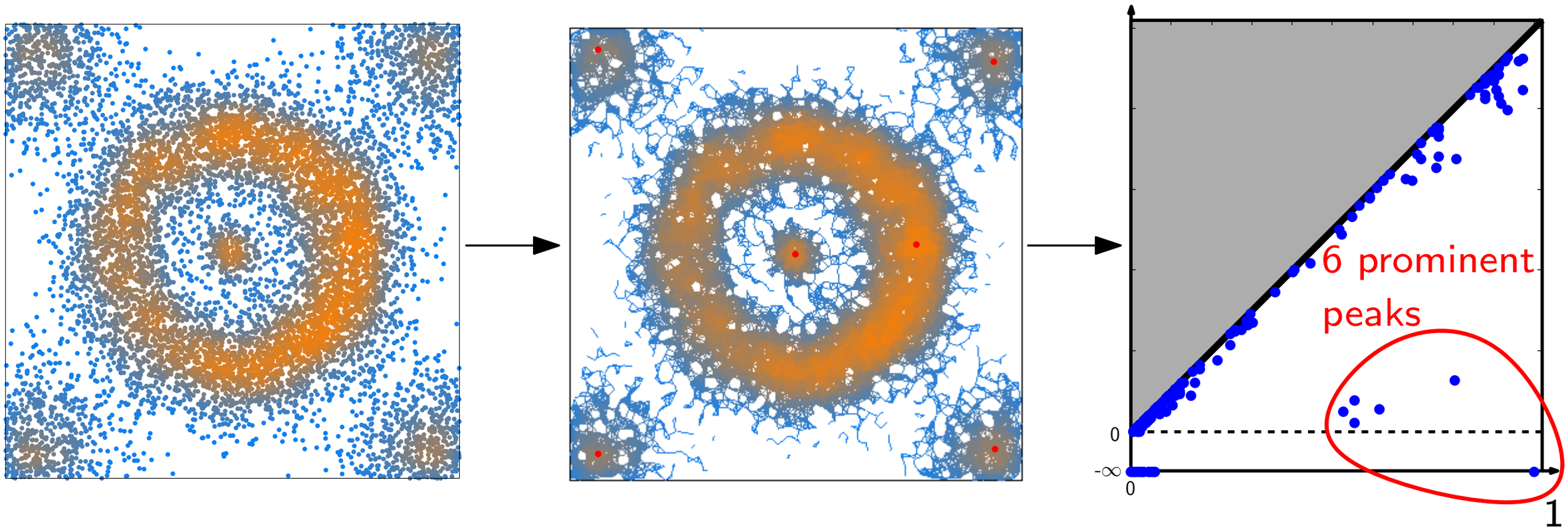
Estimating the Correct Number of Clusters



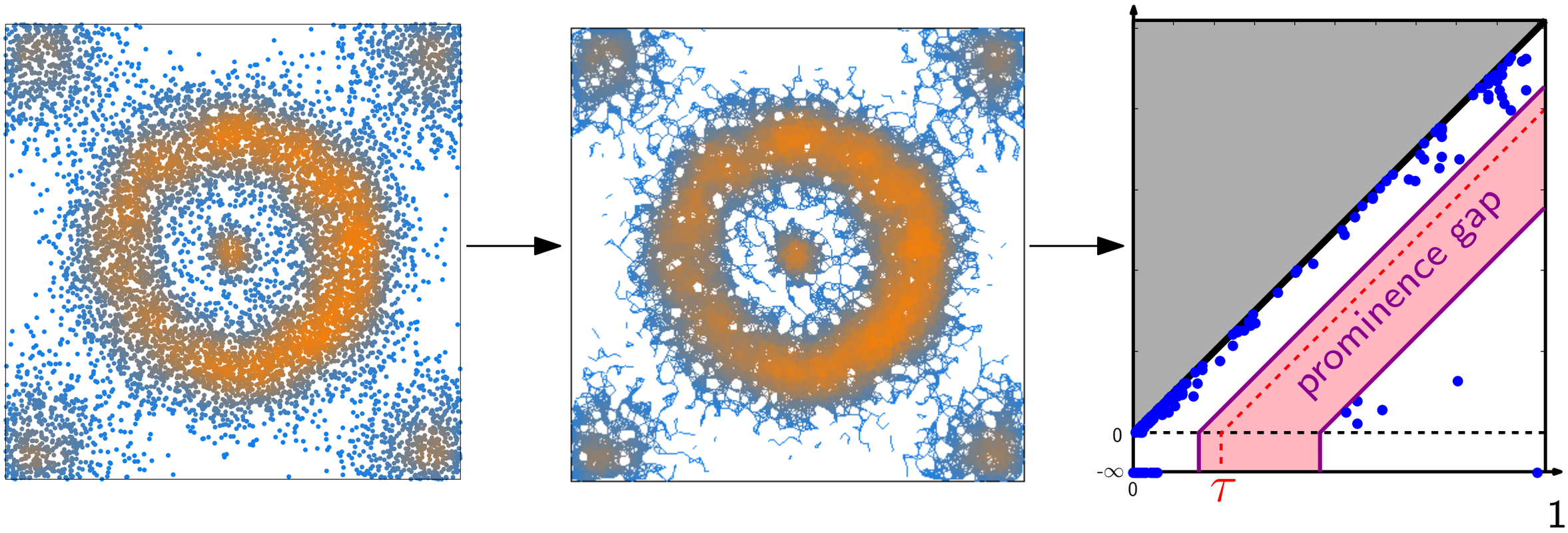
Estimating the Correct Number of Clusters



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 $\delta < \frac{\Pi - 5\eta}{5c}$.

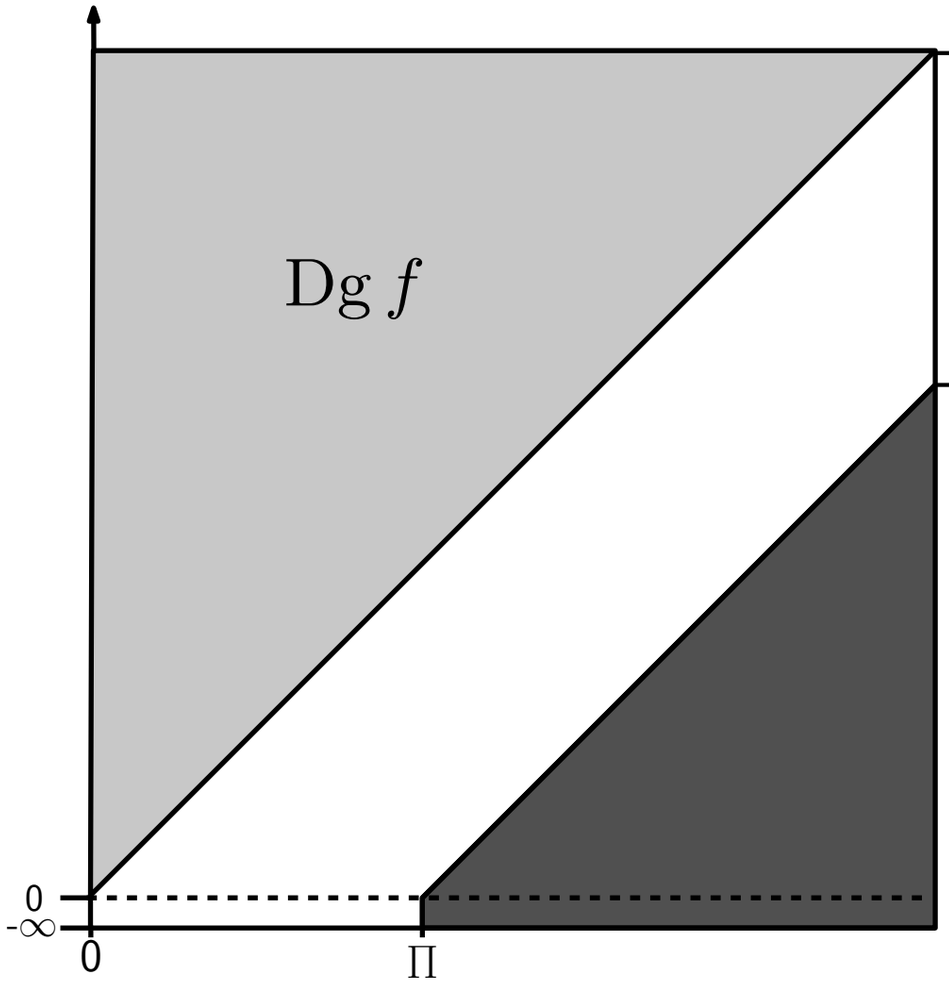
Note: Π is the prominence of the least prominent peak of f

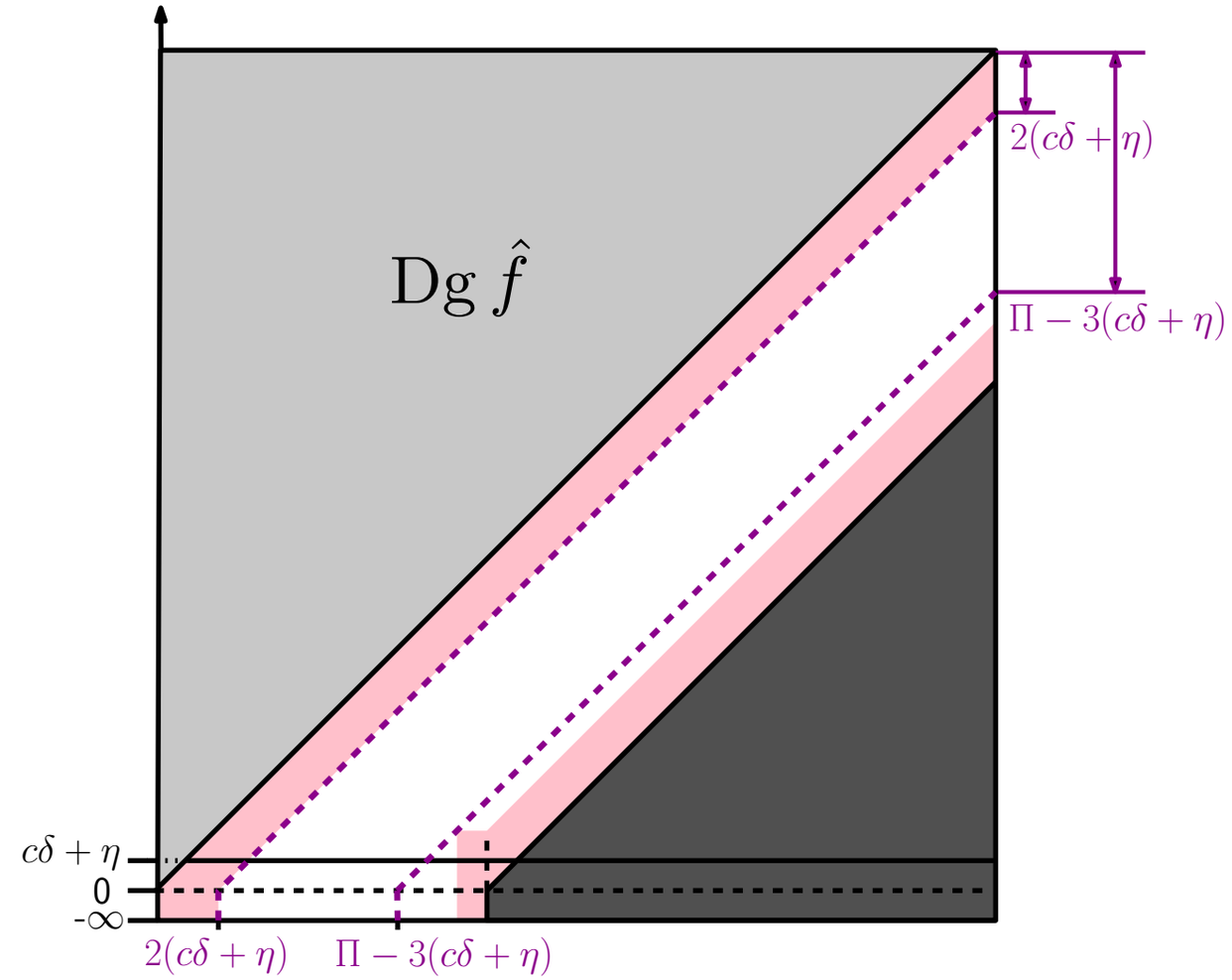
Conclusion:

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





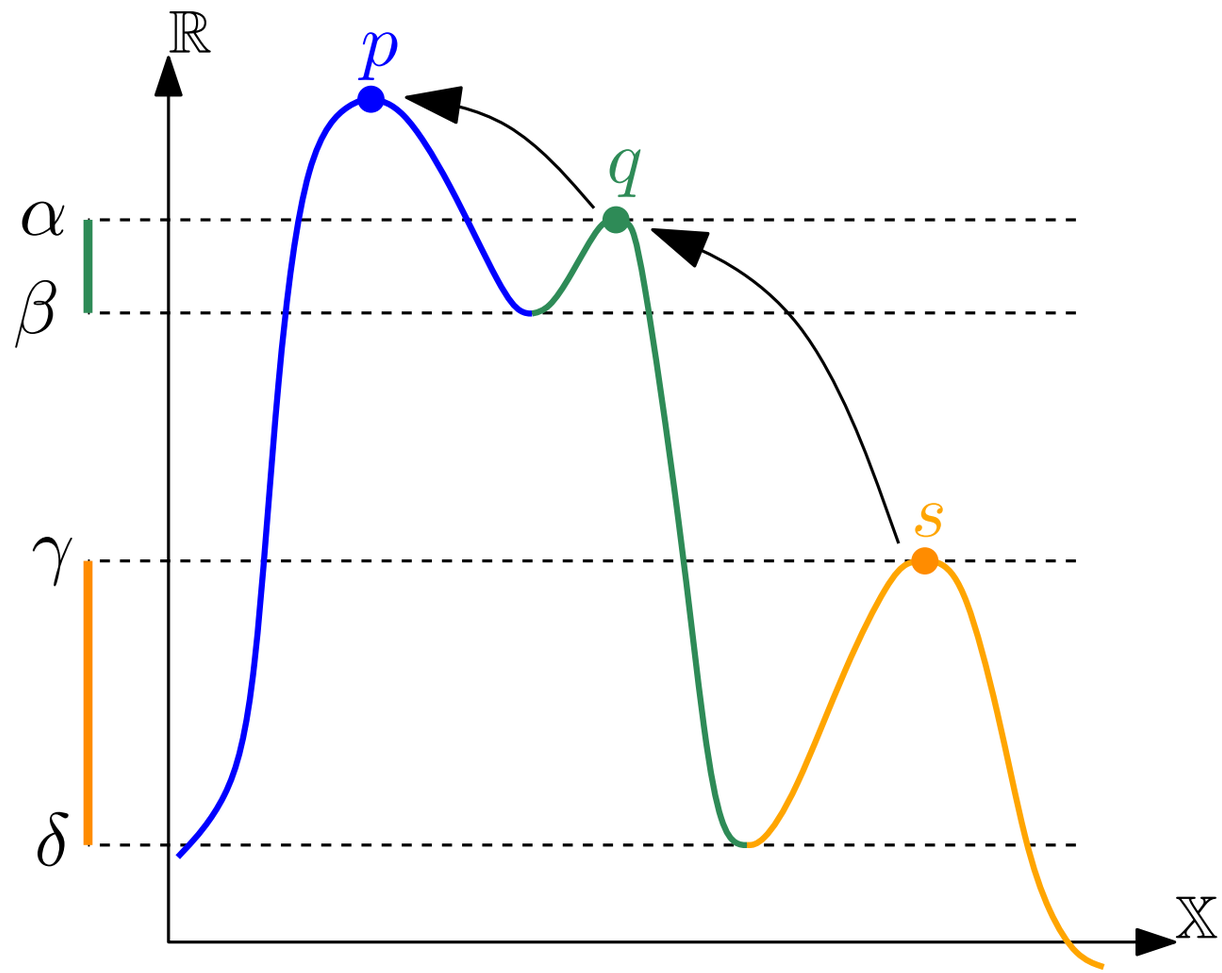
Conclusion:

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)

Merging Clusters

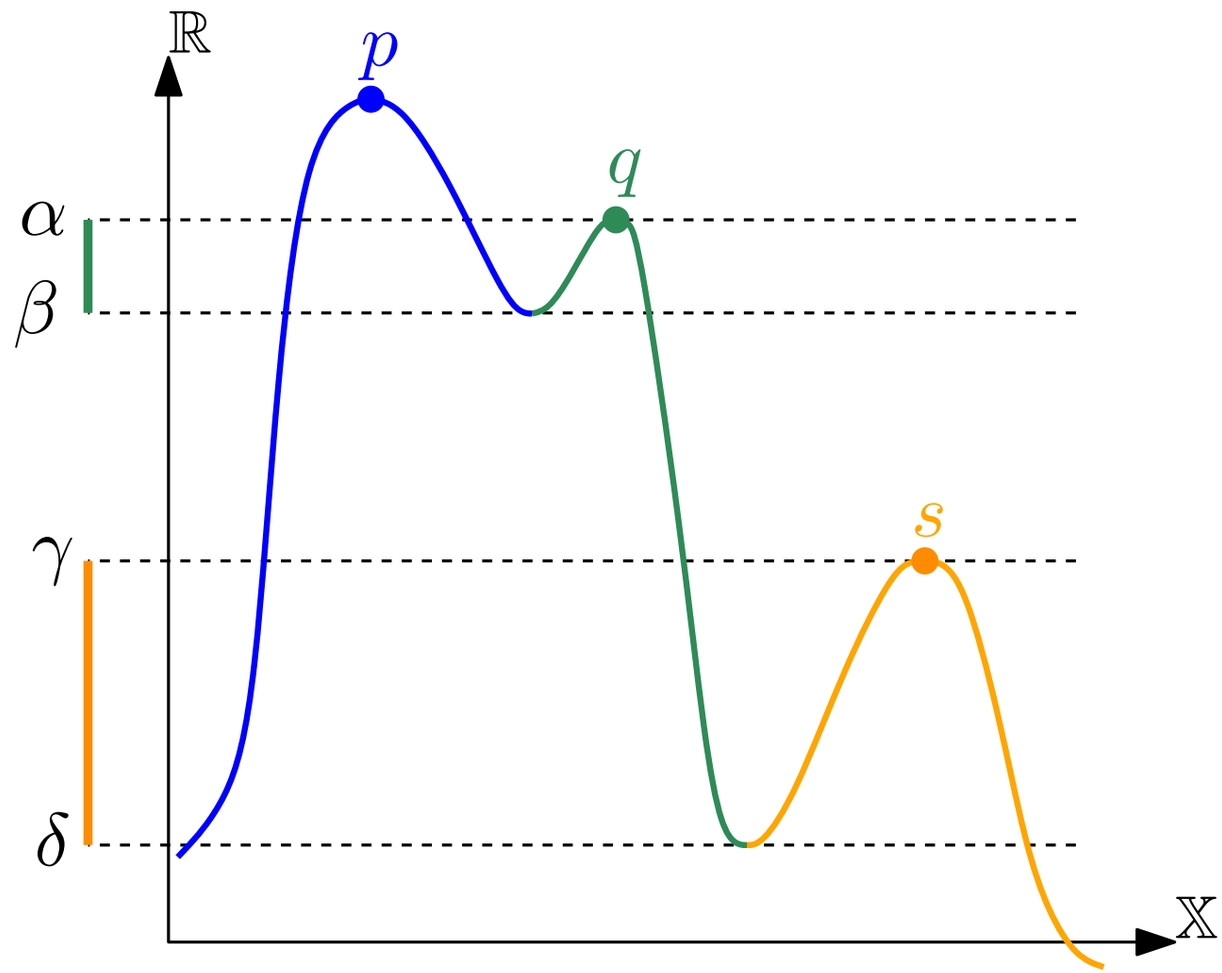
- degree-0 persistence algo. builds a hierarchy of the peaks of $\hat{f}$ (merge tree)
- merge clusters according to the hierarchy (merge each cluster into its parent)



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 $< \tau$

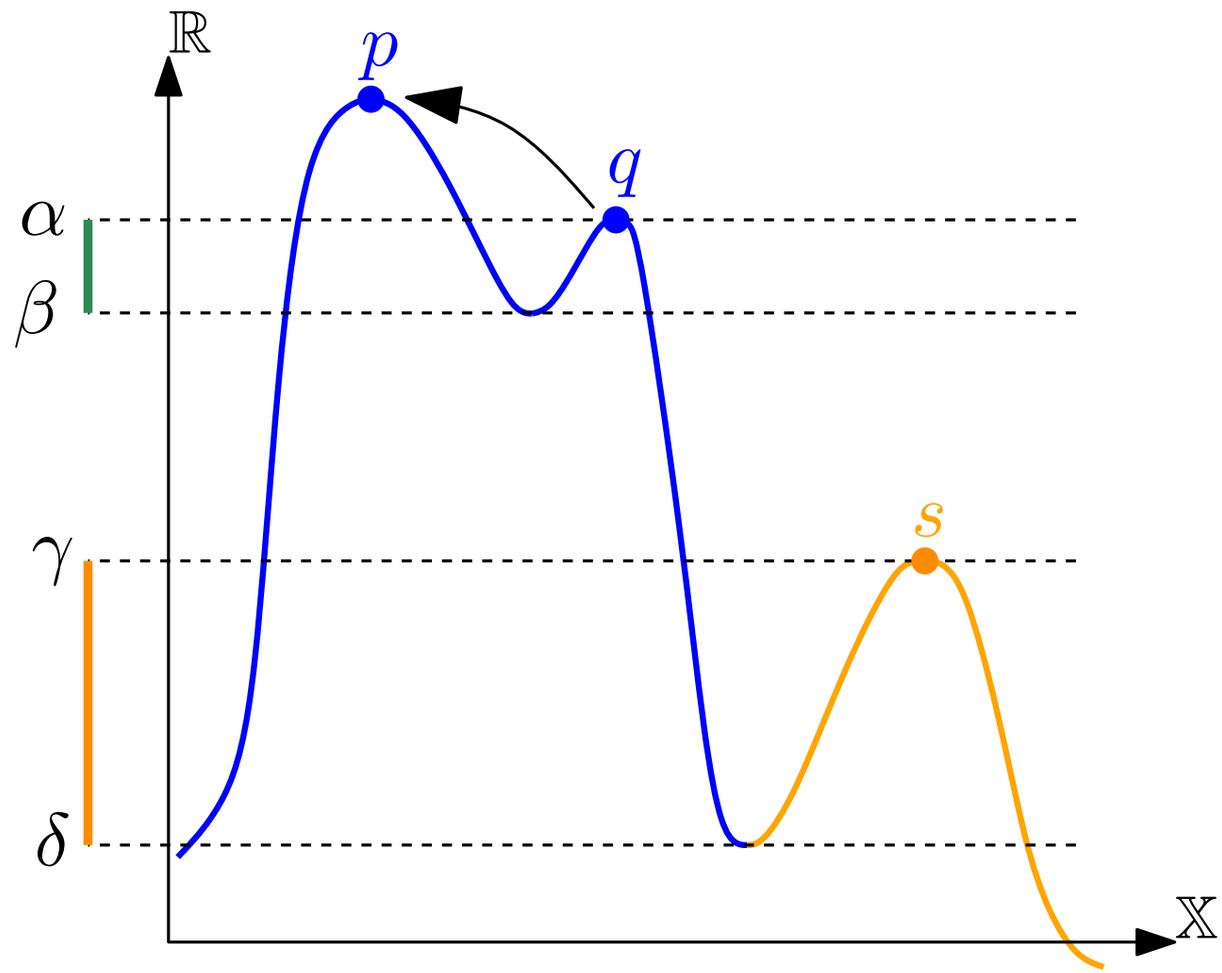
$$0 \leq \tau \leq \alpha - \beta$$



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 $< \tau$

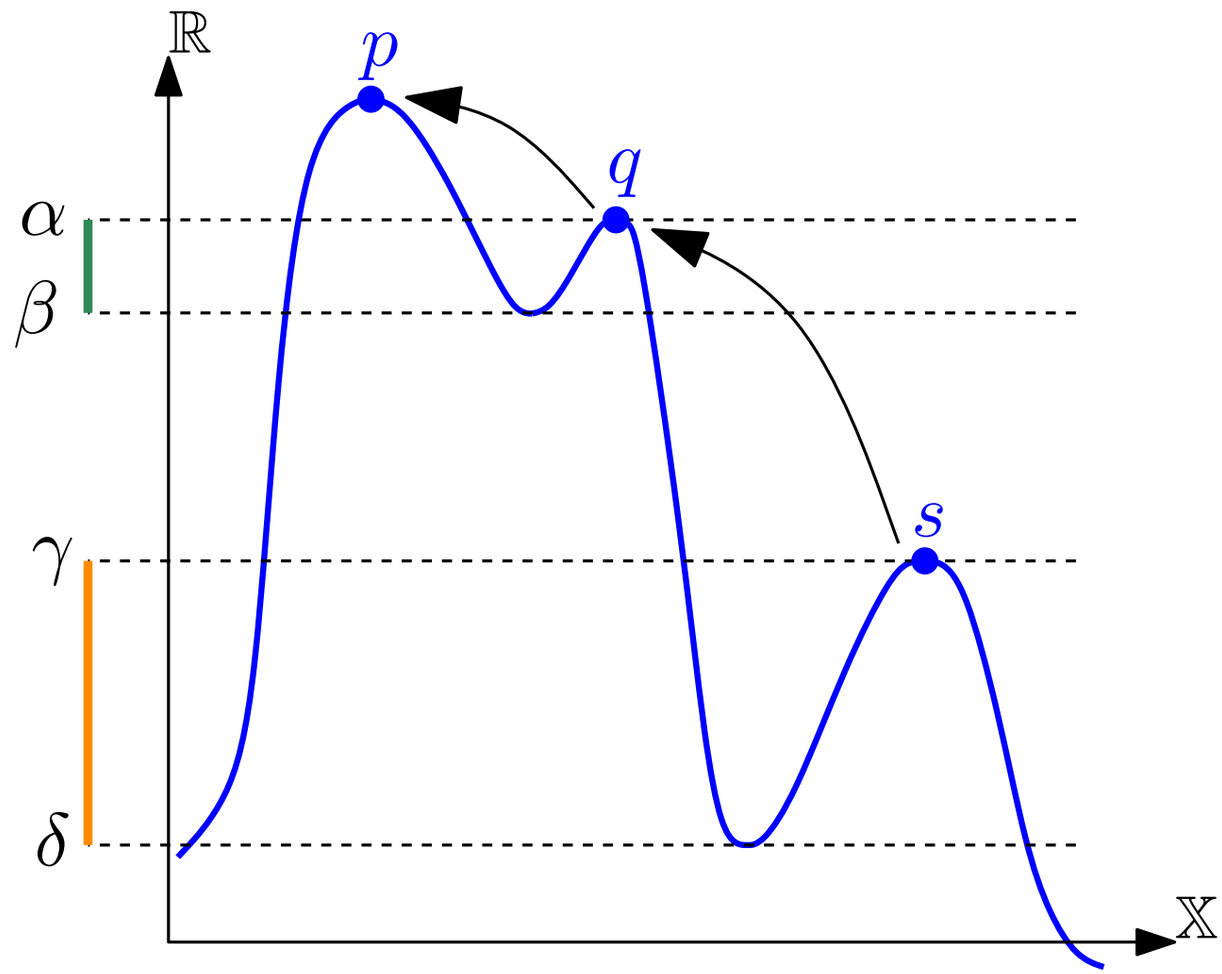
$$\alpha - \beta < \tau \leq \gamma - \delta$$



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*
        $g(i) \leftarrow \operatorname{argmax}_{j \in \mathcal{N}} \hat{f}(j)$ *// $g(i)$ stores the approximate gradient at vertex i*
        $e_i \leftarrow \mathcal{U}.\texttt{find}(g(i))$;
        Attach vertex i to the entry e_i;
    for $j \in \mathcal{N}$ **do**
        $e \leftarrow \mathcal{U}.\texttt{find}(j)$;
        if $e \neq e_i$ and $\min\{\hat{f}(r(e)),\ \hat{f}(r(e_i))\} < \hat{f}(i) + \tau$
            $\mathcal{U}.\texttt{union}(e,\ e_i)$;
            $r(e \cup e_i) \leftarrow \operatorname{argmax}_{\{r(e),\ r(e_i)\}} \hat{f}$;
            $e_i \leftarrow e \cup e_i$;

graph-based hill-climbing (1976)

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.

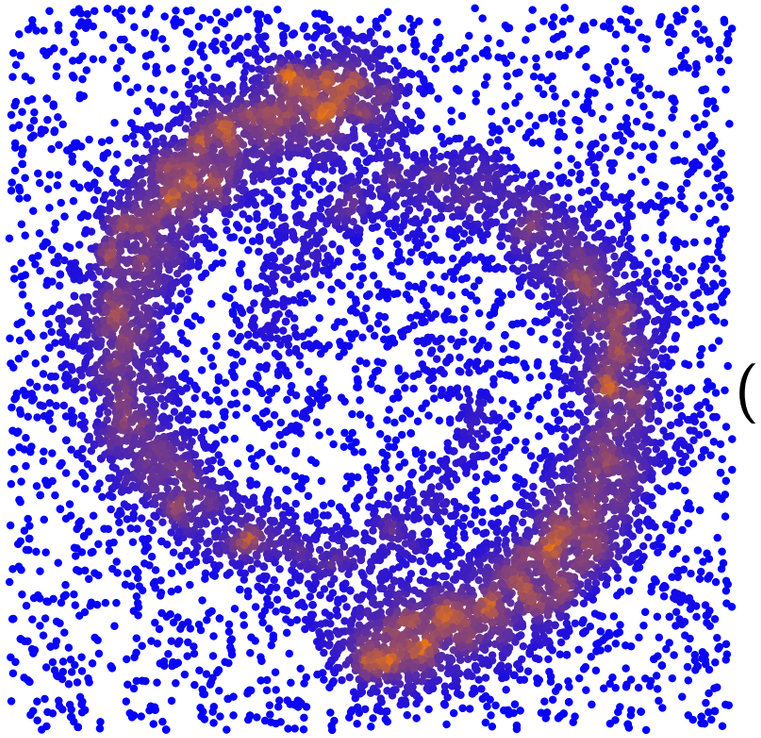
$\rightarrow$ Running time: $O(n \log n + (n + m)\alpha(n))$

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

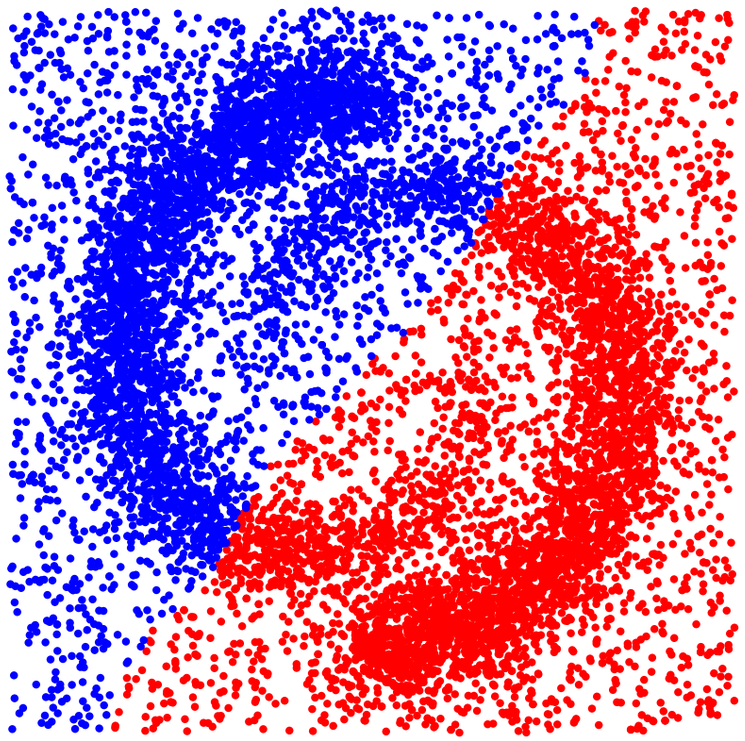
$\rightarrow$ Main memory usage: $O(n)$

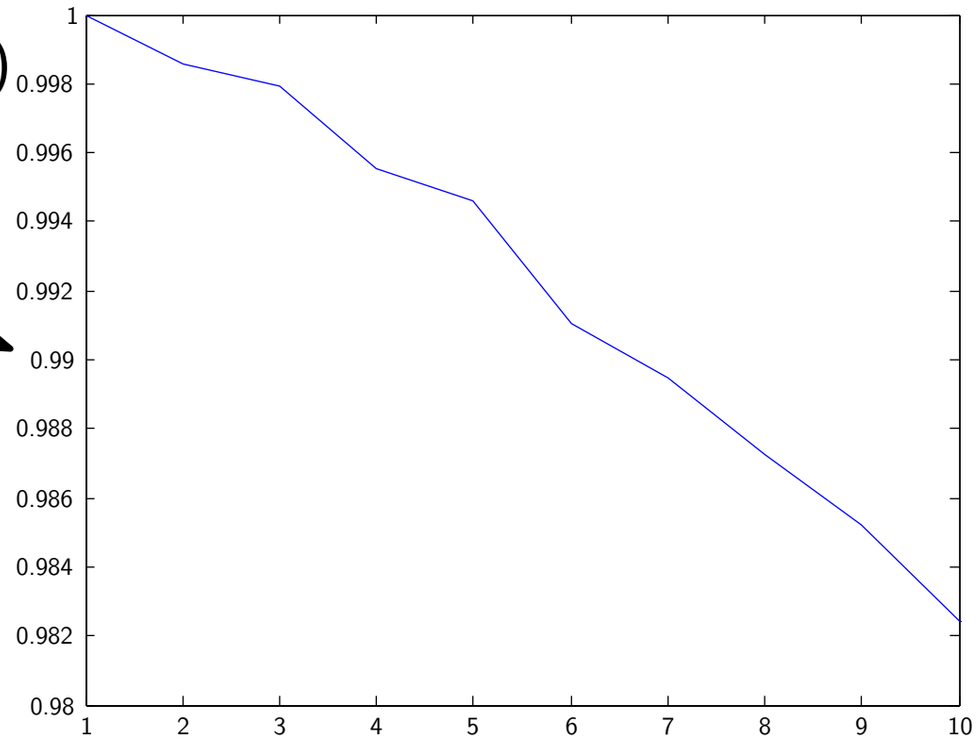
Experimental Results

Synthetic Data



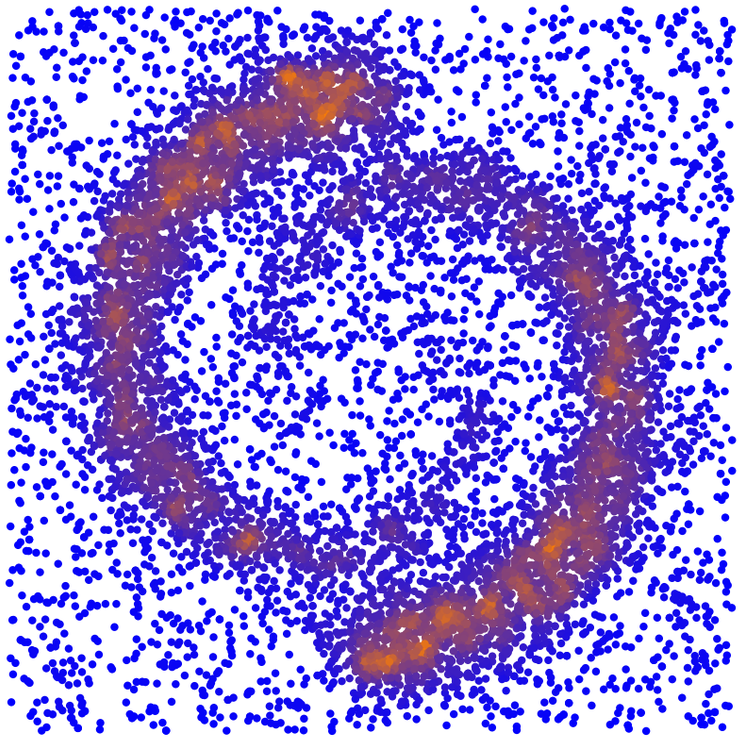
Spectral clustering
(k-means in eigenspace)





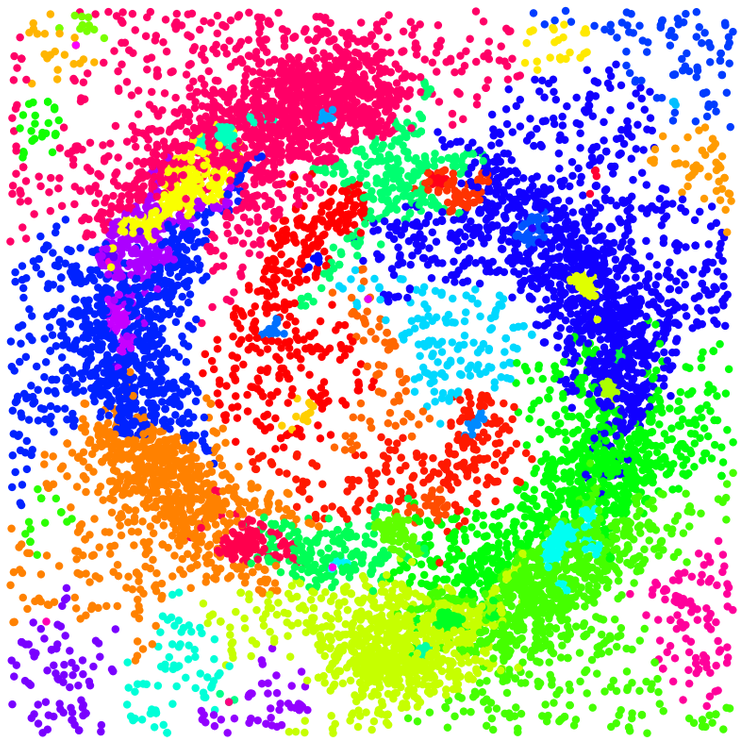
Experimental Results

Synthetic Data



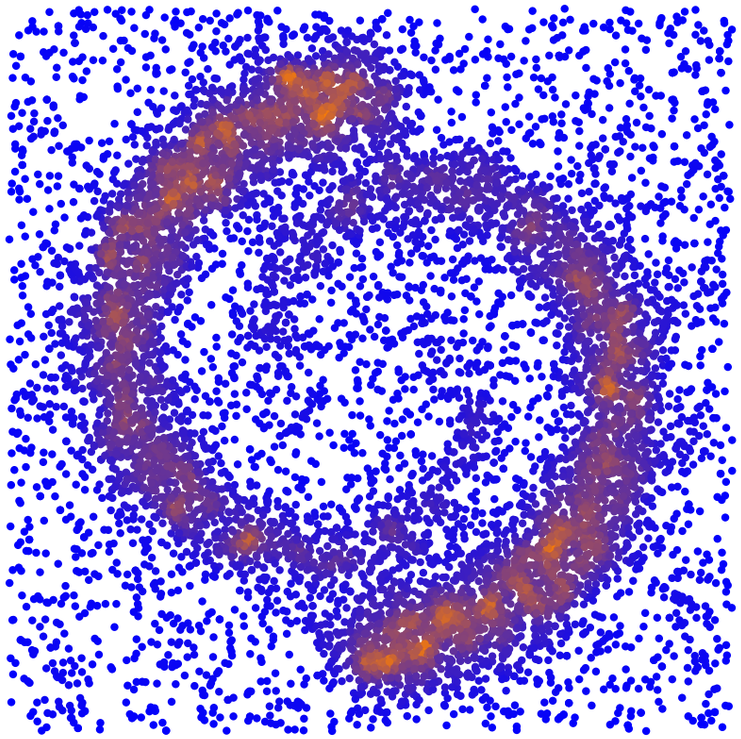
$\tau = 0$

ToMATo

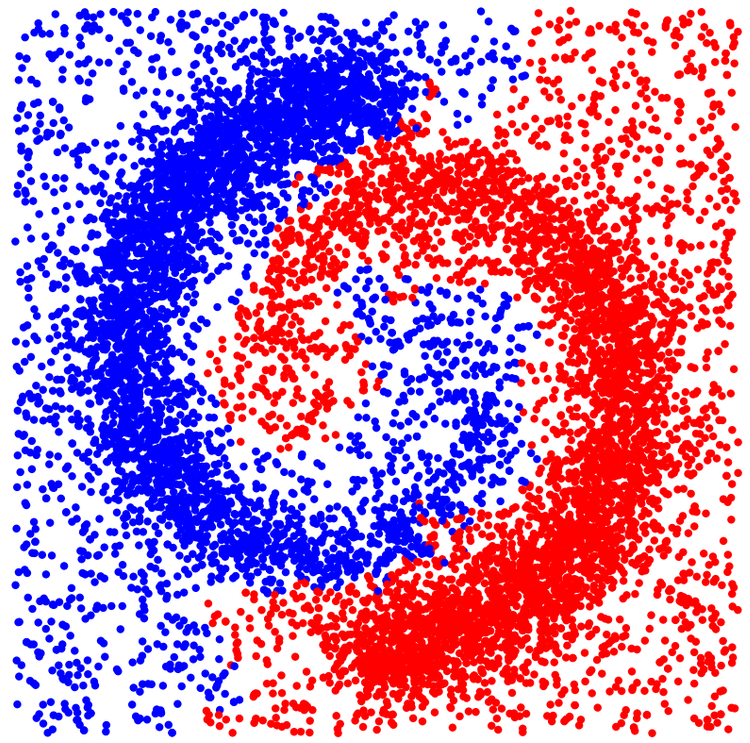


Experimental Results

Synthetic Data

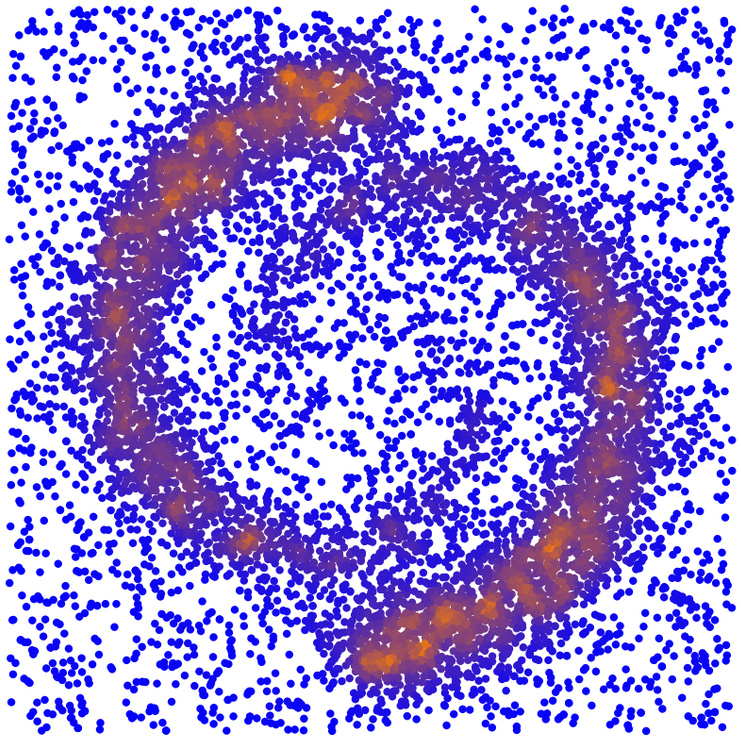


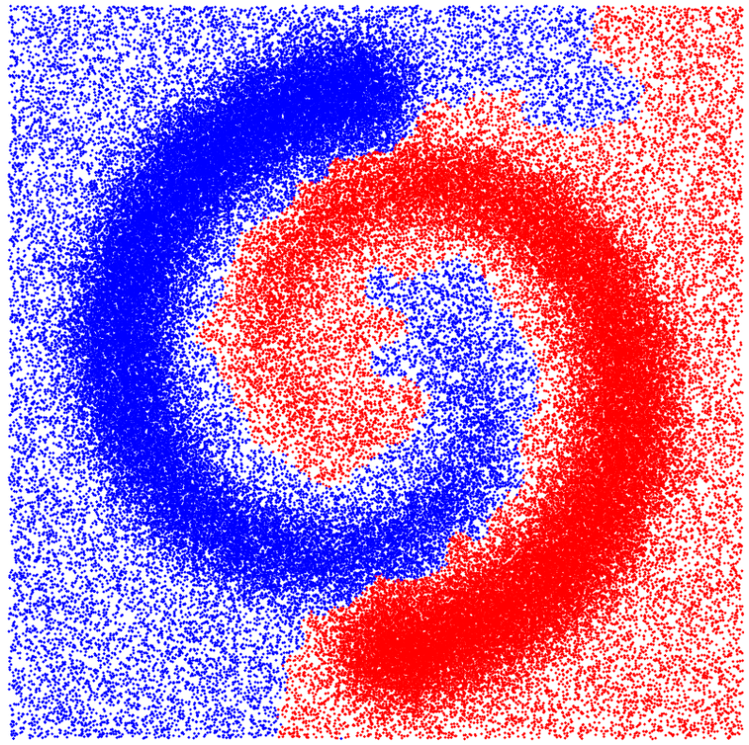
ToMATo

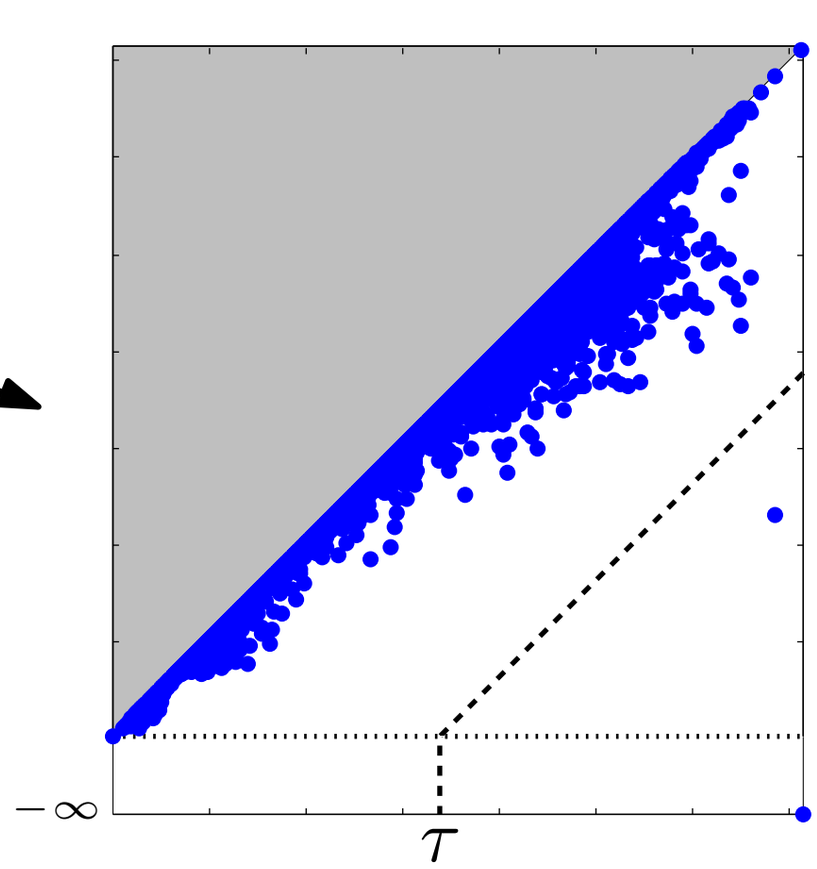


Experimental Results

Synthetic Data





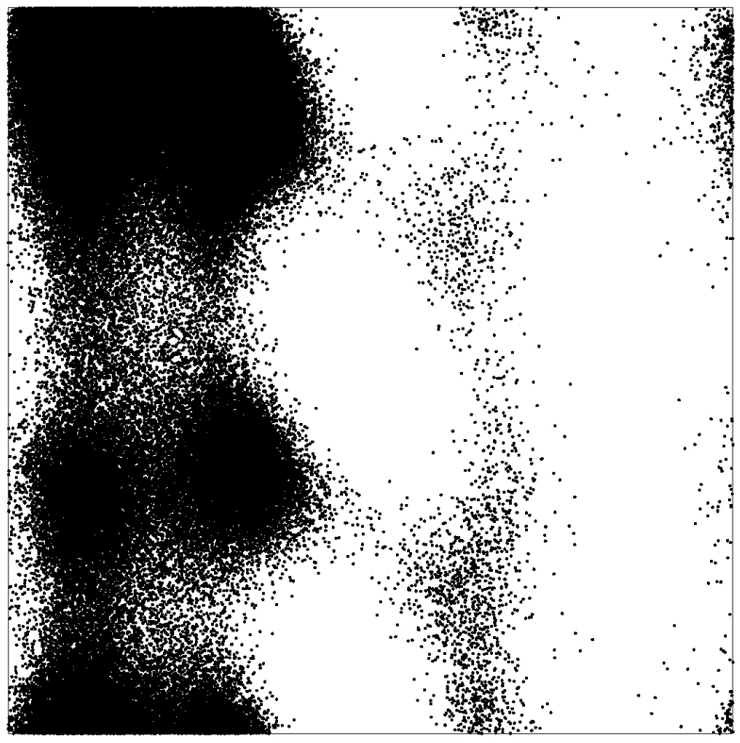


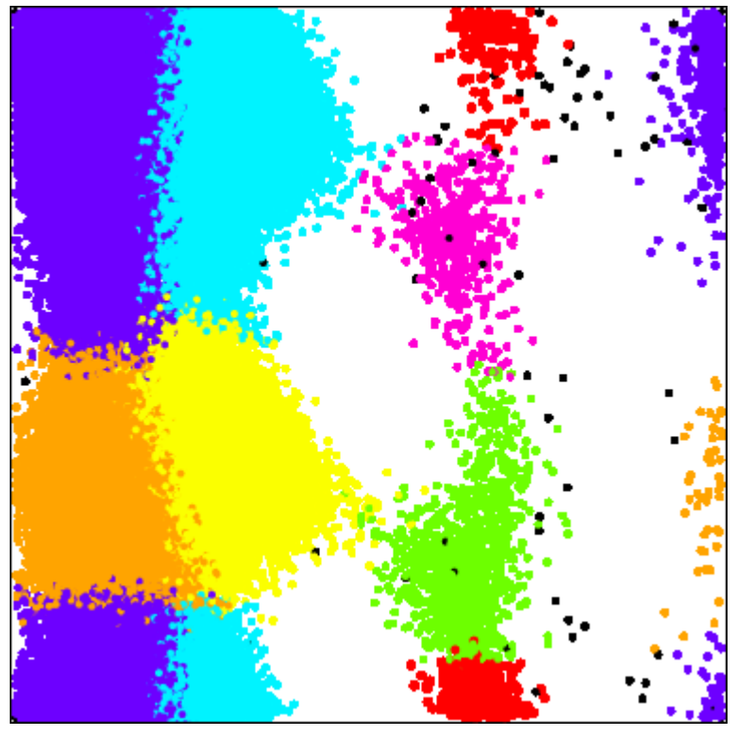
Experimental Results

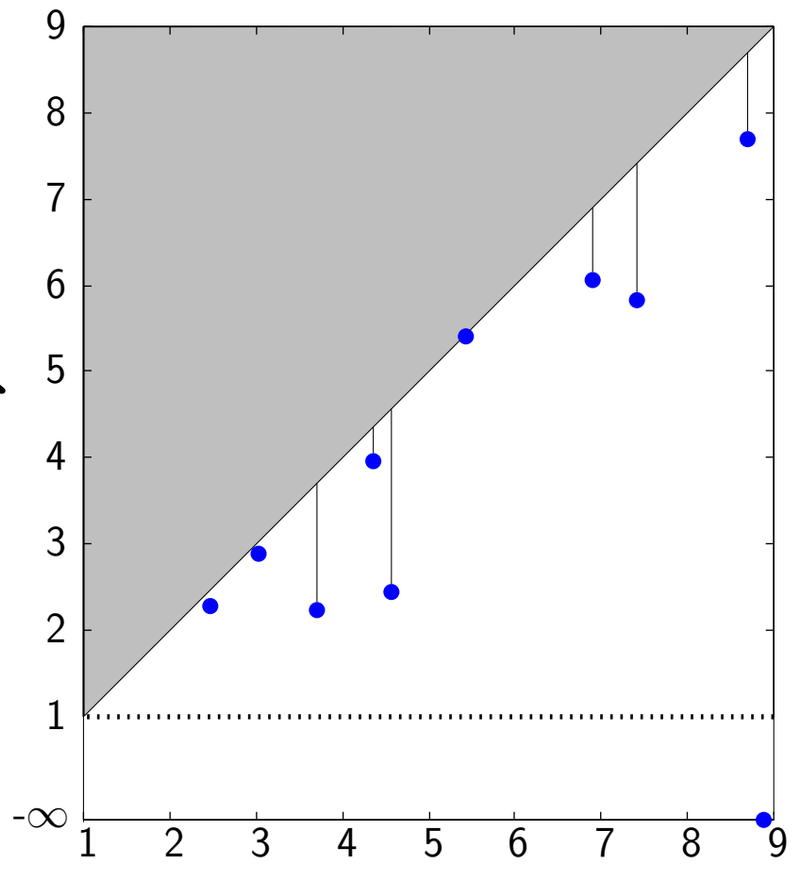
Biological Data

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

RMSD distance (non-Euclidean)







Common belief: 6 metastable states

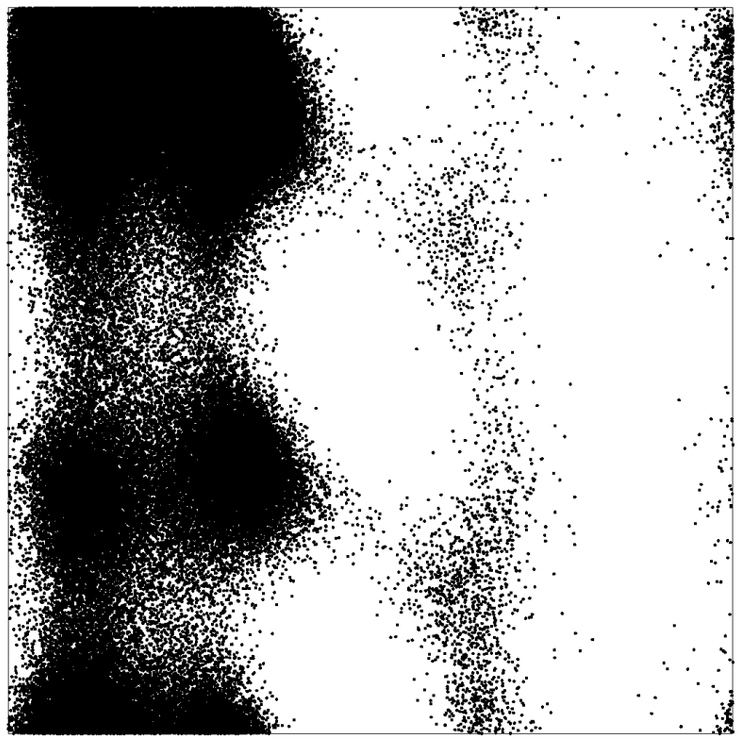
PD shows anywhere between 4 and 7 clusters

Experimental Results

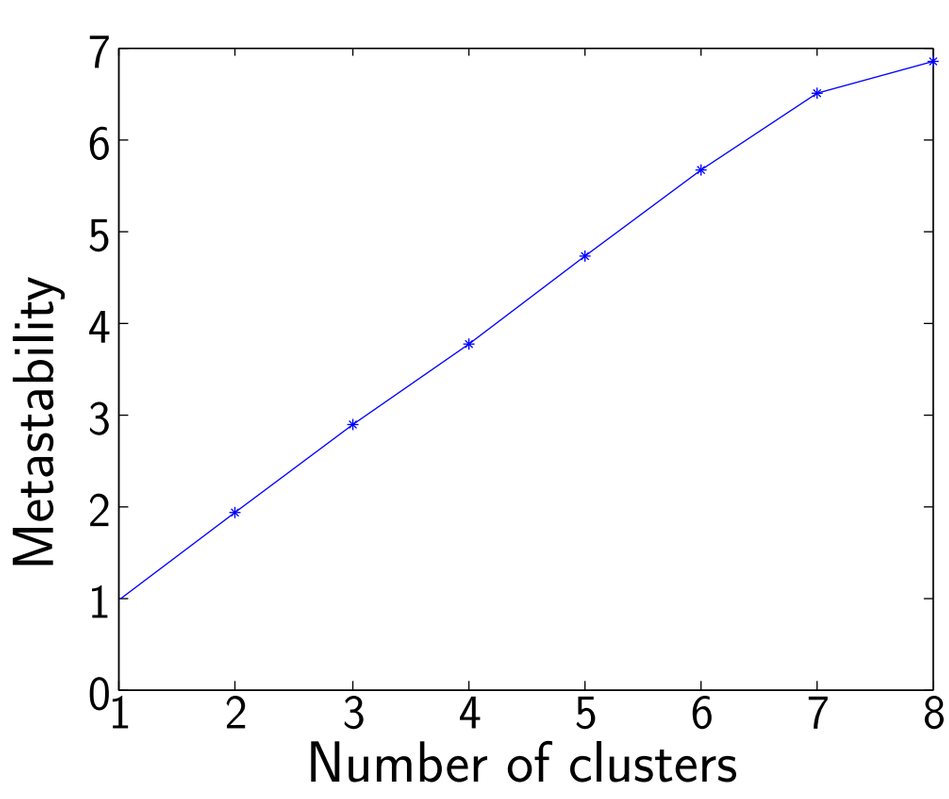
Biological Data

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

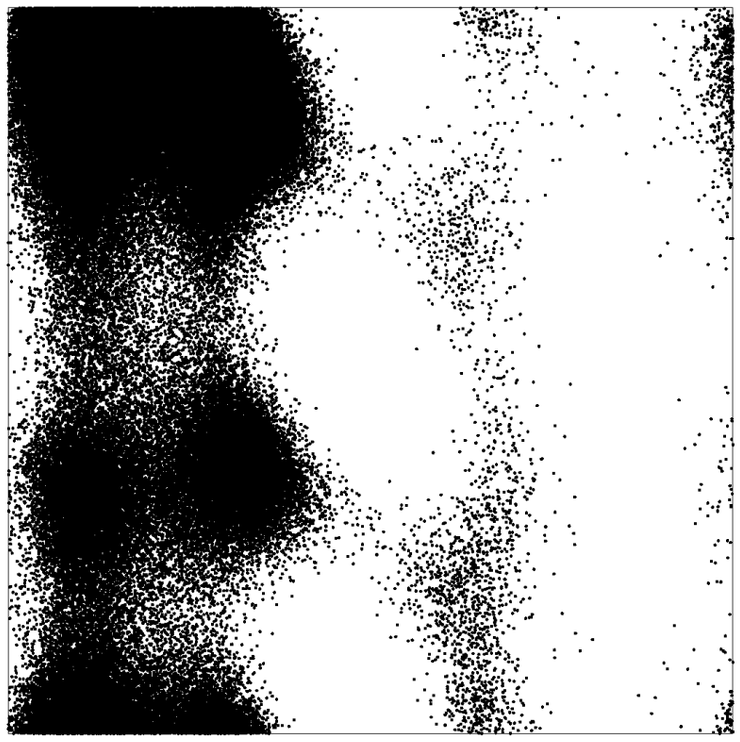
Measures of metastability confirm this insight

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.

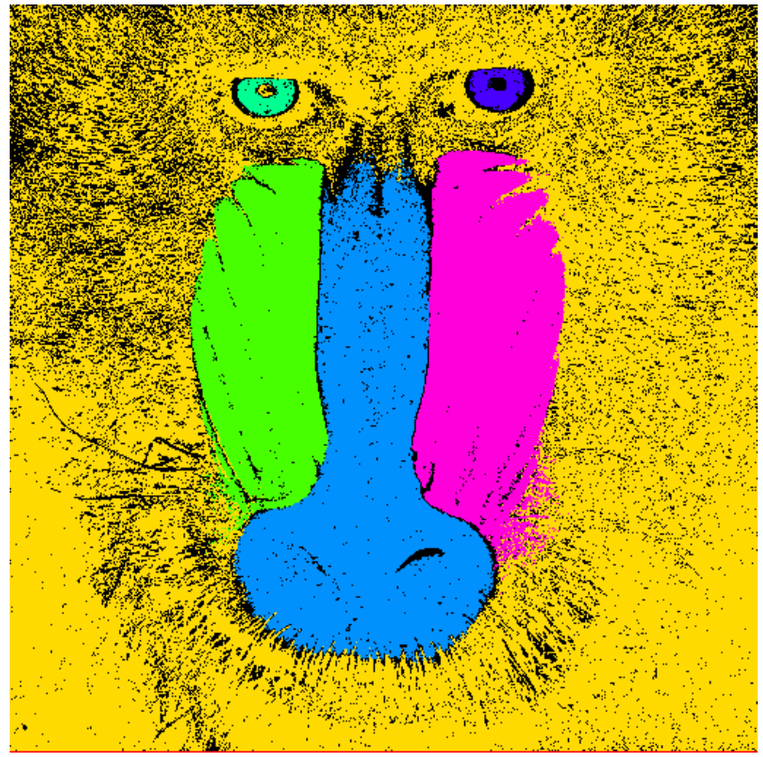
Experimental Results

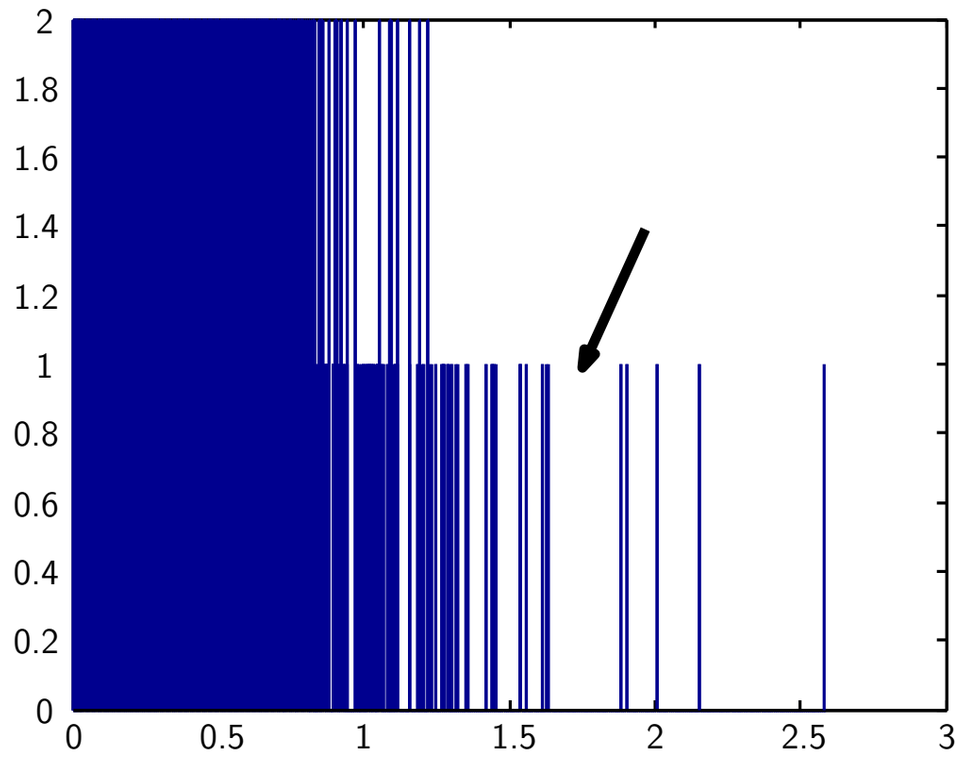
Image Segmentation

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