Spectral Clustering for Detecting Stationary Subregions in Gaussian Process Regression

Théo SYLVESTRE (CEA/DM2S/SGLS/LIAD, LMA Avignon)

PhD director: Amandine MARREL (CEA Cadarache, LMA)

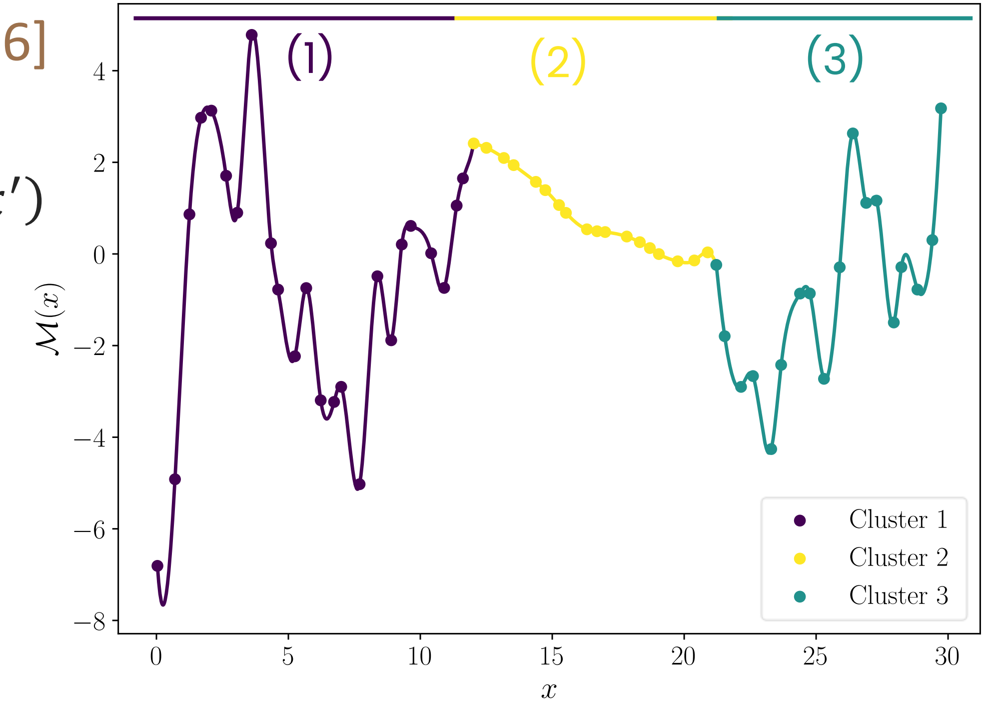
Supervisors: Guillaume DAMBLIN, Raksmy NOP (CEA Paris-Saclay)

Laboratory: LIAD/ LMA
Doctoral school: ED 536

Context

Setting:

- Surrogate modelling for expensive computer codes $\mathcal{M}: x \in \mathbb{R}^d \mapsto \mathcal{M}(x) \in \mathbb{R}$
- Popular solution: Gaussian Process ($\mathcal{GP}$) surrogate $\widehat{\mathcal{M}}$ [RW06]
  - ➢ $\mathcal{GP}$ characterised by a mean $m(\cdot)$ and covariance $k(.,.)$
  - ➢ Standard assumption: stationary covariance $k(x, x') = k(x - x')$



Non-stationary fonction with 3 stationary subregions

[RW06] Rasmussen, C.E., Williams, C.K.I. (2006). Gaussian processes for machine learning. MIT Press.

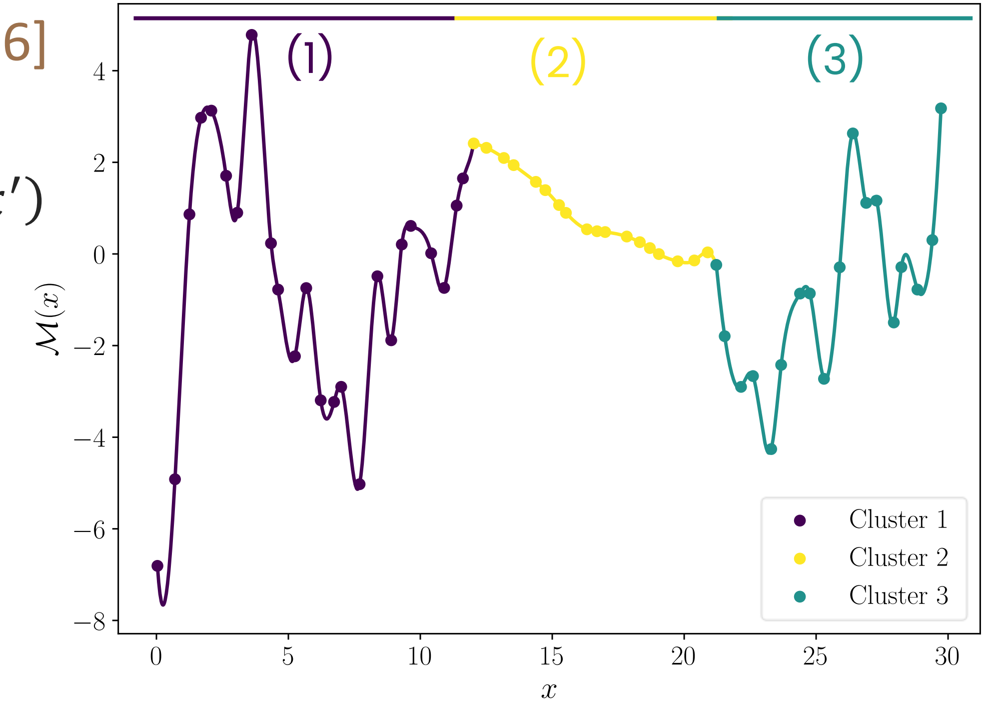
Context

Setting:

- Surrogate modelling for expensive computer codes $\mathcal{M}: x \in \mathbb{R}^d \mapsto \mathcal{M}(x) \in \mathbb{R}$
- Popular solution: Gaussian Process ($\mathcal{GP}$) surrogate $\widehat{\mathcal{M}}$ [RW06]
  - ➢ $\mathcal{GP}$ characterised by a mean $m(\cdot)$ and covariance $k(.,.)$
  - ➢ Standard assumption: stationary covariance $k(x, x') = k(x - x')$

Limitation:

- Stationarity assumption inadequate when $\mathcal{M}$ exhibits distinct regimes
- Ignoring non-stationarity → innacurate $\mathcal{GP}$ predictions and poorly calibrated predictive uncertainties



Non-stationary fonction with 3 stationary subregions

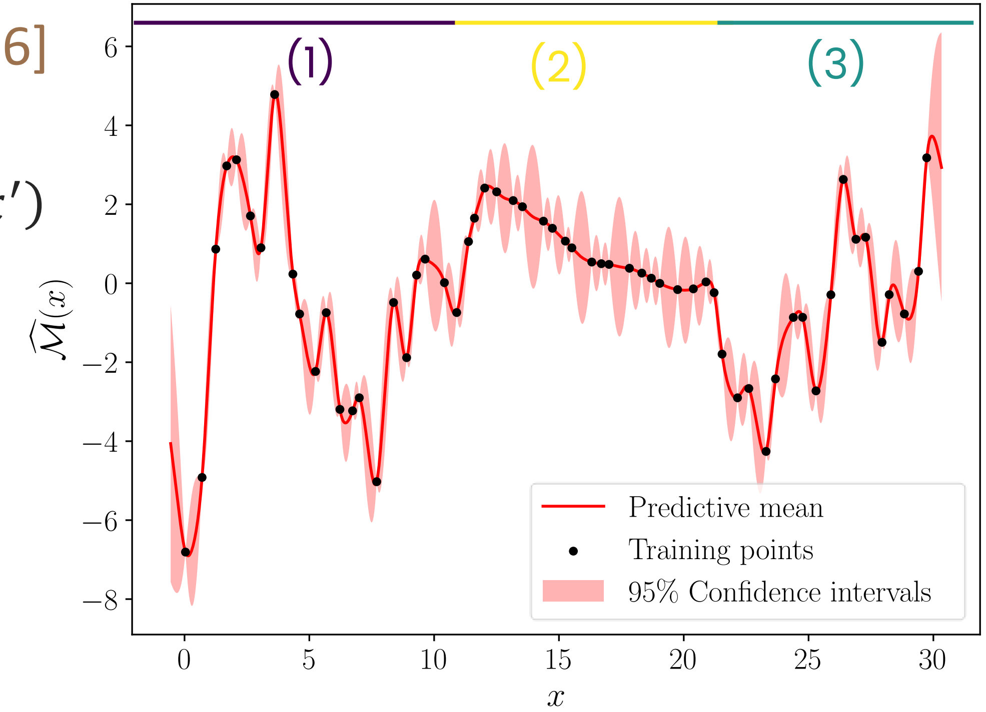
Context

Setting:

- Surrogate modelling for expensive computer codes $\mathcal{M}: x \in \mathbb{R}^d \mapsto \mathcal{M}(x) \in \mathbb{R}$
- Popular solution: Gaussian Process ($\mathcal{GP}$) surrogate $\widehat{\mathcal{M}}$ [RW06]
  - $\mathcal{GP}$ characterised by a mean $m(\cdot)$ and covariance $k(.,.)$
  - Standard assumption: stationary covariance $k(x,x') = k(x - x')$

Limitation:

- Stationarity assumption inadequate when $\mathcal{M}$ exhibits distinct regimes
- Ignoring non-stationarity → innacurate $\mathcal{GP}$ predictions and poorly calibrated predictive uncertainties



Stationary $\mathcal{GP}$ on the non-stationary function

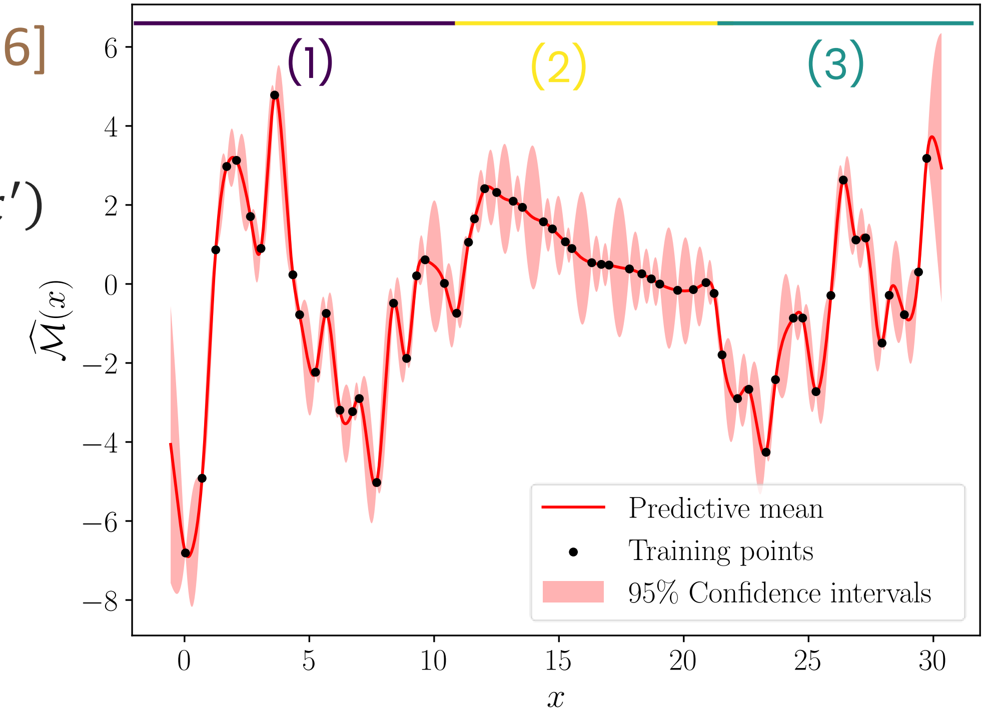
Context

Setting:

- Surrogate modelling for expensive computer codes $\mathcal{M}: x \in \mathbb{R}^d \mapsto \mathcal{M}(x) \in \mathbb{R}$
- Popular solution: Gaussian Process ($\mathcal{GP}$) surrogate $\widehat{\mathcal{M}}$ [RW06]
  - ➢ $\mathcal{GP}$ characterised by a mean $m(\cdot)$ and covariance $k(.,.)$
  - ➢ Standard assumption: stationary covariance $k(x, x') = k(x - x')$

Limitation:

- Stationarity assumption inadequate when $\mathcal{M}$ exhibits distinct regimes
- Ignoring non-stationarity → innacurate $\mathcal{GP}$ predictions and poorly calibrated predictive uncertainties



Stationary $\mathcal{GP}$ on the non-stationary function

Goal:

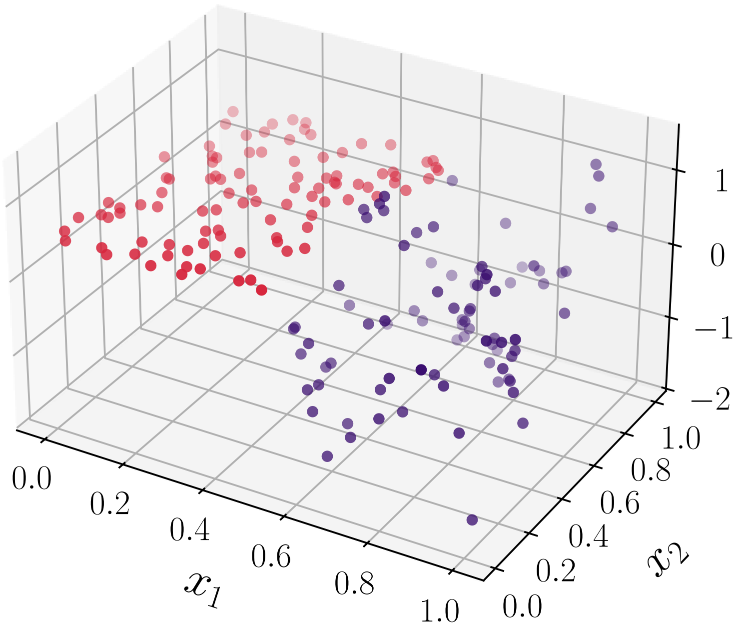
- Need to build a tailored surrogate
  - ➢ Detect stationary regimes (subregions) and combine local stationary $\mathcal{GP}$ surrogates ($\mathcal{GP}$ mixture)

[RW06] Rasmussen, C.E., Williams, C.K.I. (2006). Gaussian processes for machine learning. MIT Press.

General approach

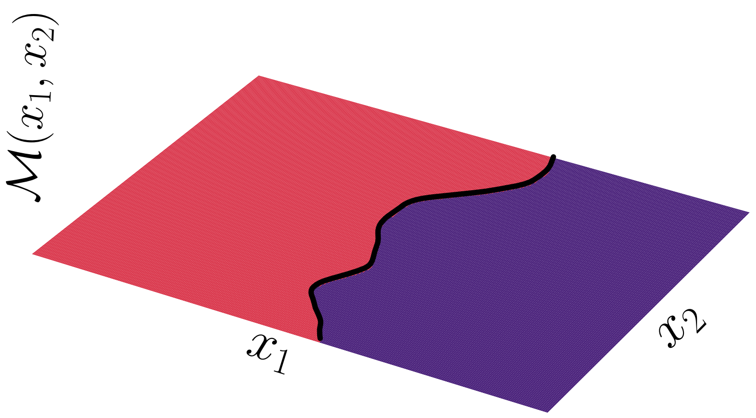
- Three-step approach adapted from [MS23]

<u>Clustering</u>



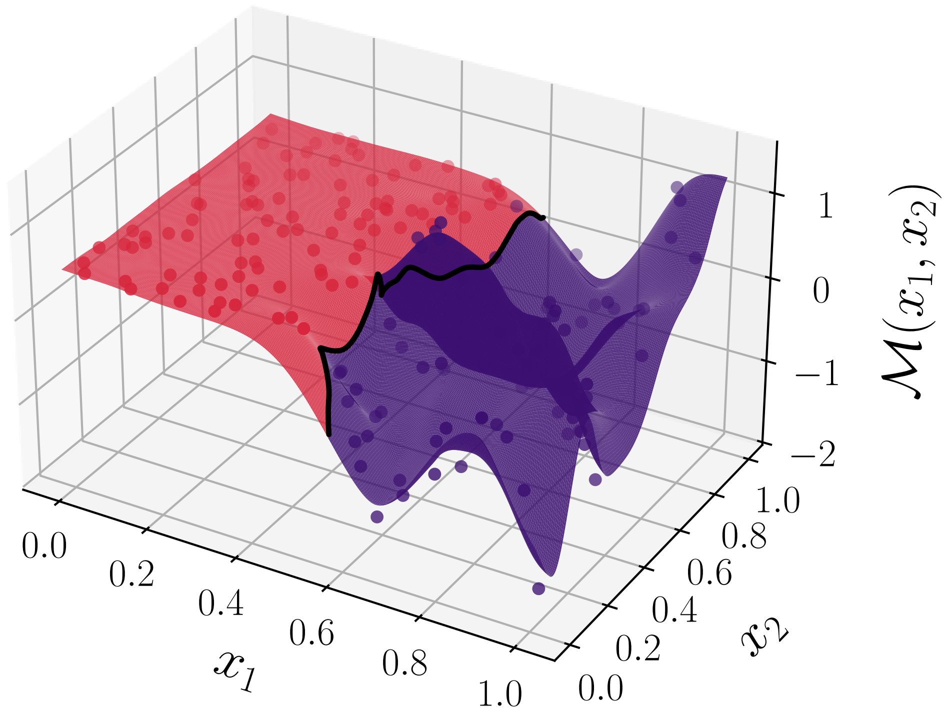
Detection of stationary subregions (regimes) using adapted features

<u>Classification</u>



Estimate, for any new input, its probability of belonging to each regime

<u>Regression</u>



Combine local stationary $\mathcal{GP}$ surrogates into the final non-stationary $\mathcal{GP}$ surrogate

[MS23] Moustapha, M., Sudret, B. (2023). Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling. Computers & Structures, Elsevier.

Step 1: Clustering

Existing approaches and limitations

- Standard clustering algorithms (e.g Kmeans, Dirichlet processes,...) are directly applied to the joint input–output space $(\boldsymbol{x}, \mathcal{M}(\boldsymbol{x}))$

- Limitations and proposed solutions:

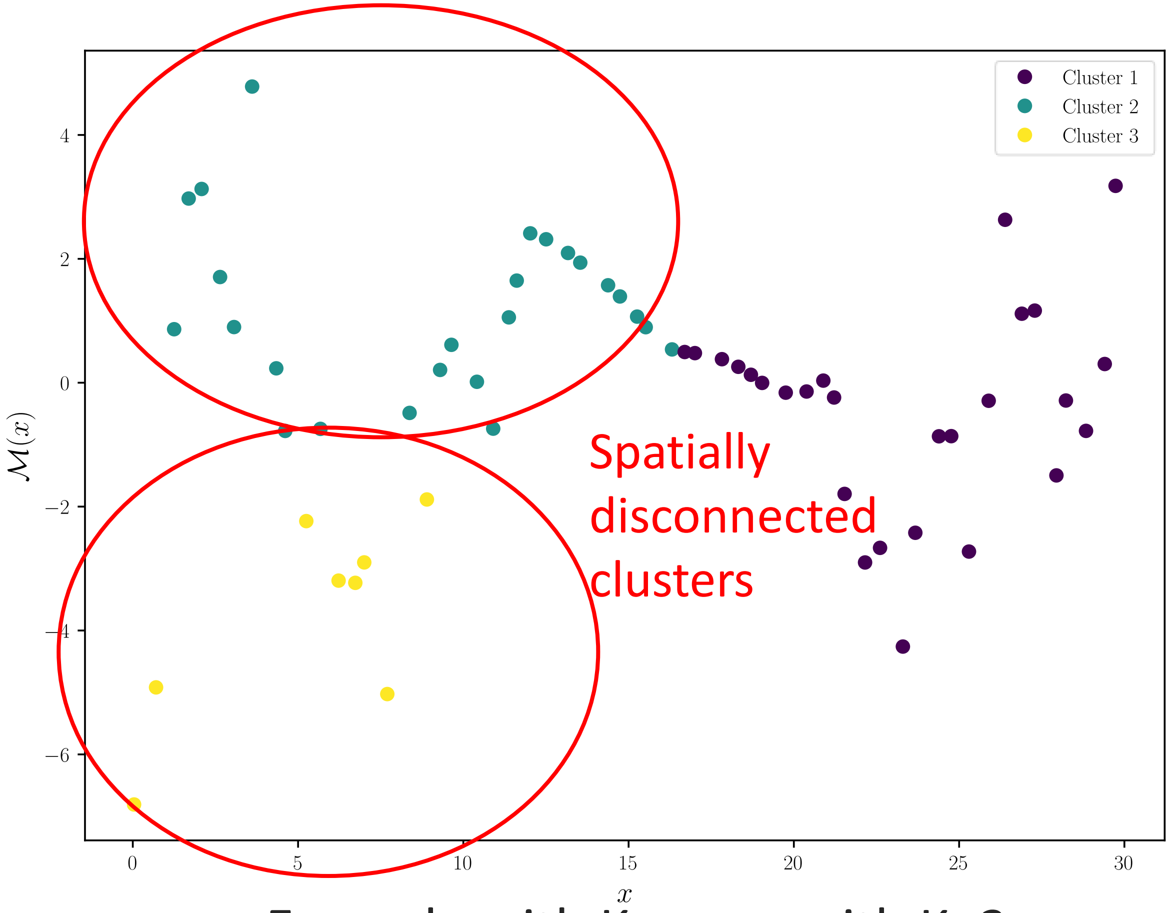
L1. Clusters may be spatially disconnected

Not suitable for $\mathcal{GP}$ models

➢ Enforce path-connected clusters via additional connectivity constraint (Voronoi tessellation)

L2. No information on stationarity is used to form the clusters

➢ Use additional local features characterising stationarity



Example with Kmeans with K=3

Local features characterising stationarity (L2)

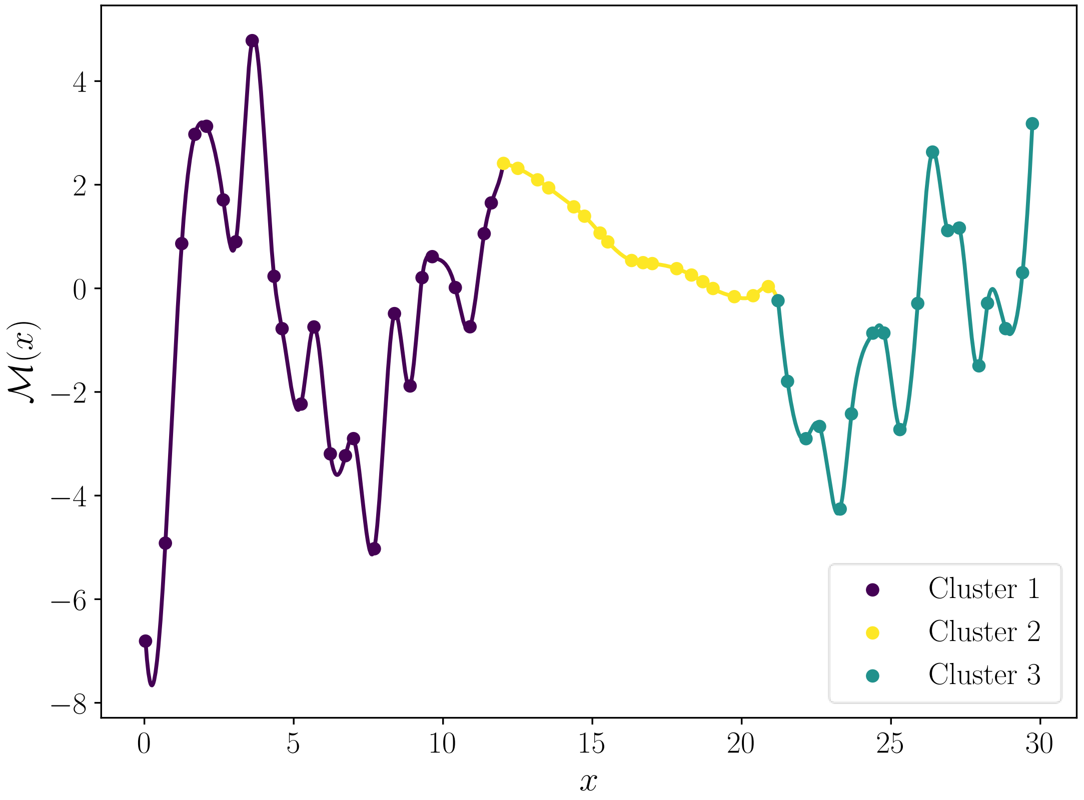
- Two points belong to the same stationary regime if they exhibit similar local behavior regarding:

  - **Input position** $\boldsymbol{x}_i$:
    Spatial coherence

  - **Local variance** $v_i = \mathrm{Var}(\mathcal{M}(\boldsymbol{x}_i)|\boldsymbol{x}_i)$:
    Local amplitude

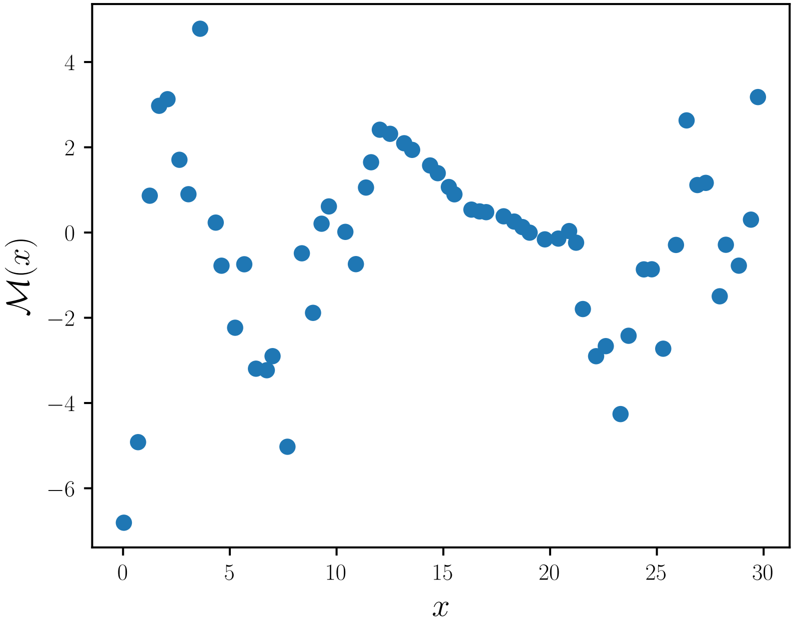
  - **Gradient** $\boldsymbol{\nabla}_i = \boldsymbol{\nabla}\mathcal{M}(\boldsymbol{x}_i)$:
    Local directional variation

- Let $\mathbf{z}_i = (\boldsymbol{x}_i, v_i, \boldsymbol{\nabla}_i)_{i\in[\![1,n]\!]}$ denote the resulting feature points



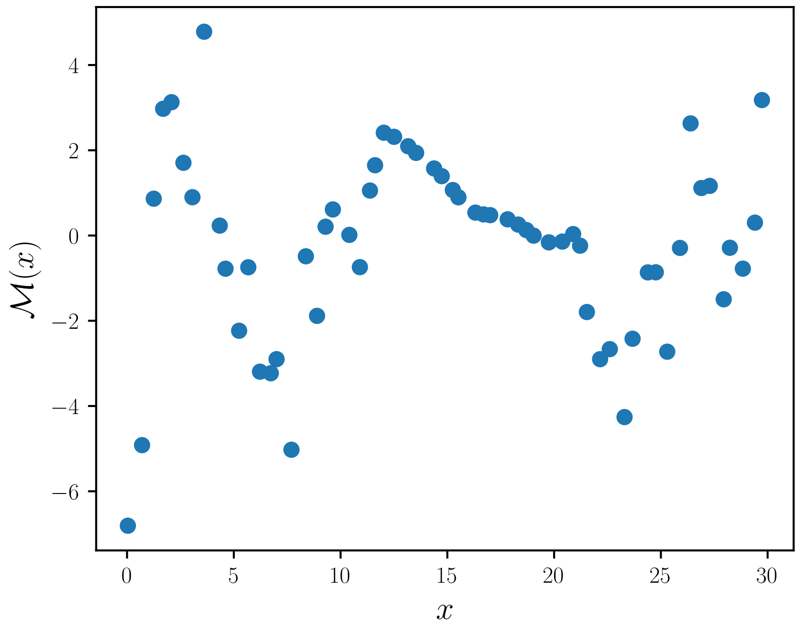
Clustering to detect stationary subregions

Observed data



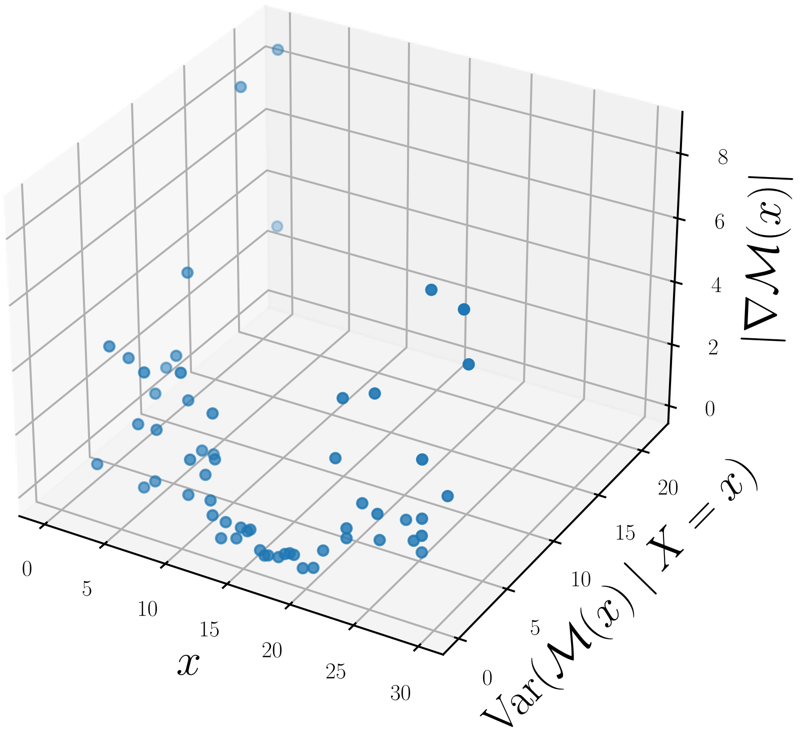
Clustering to detect stationary subregions

Observed data



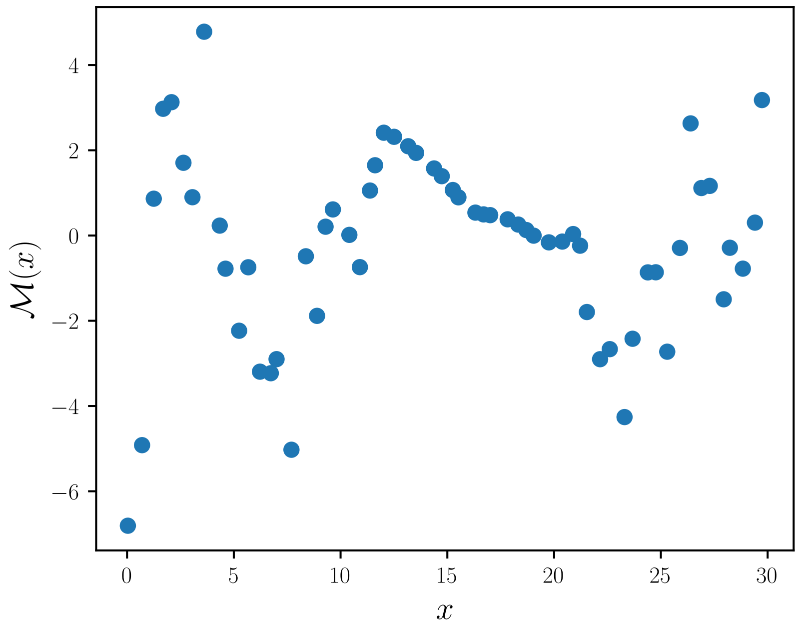
Features extraction

Feature space



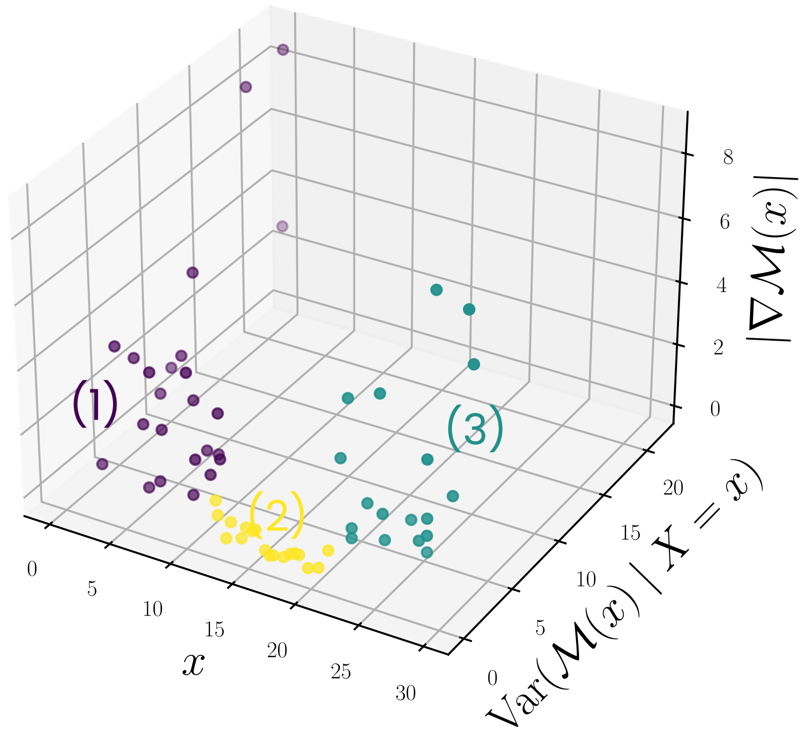
Clustering to detect stationary subregions

Observed data



Features extraction

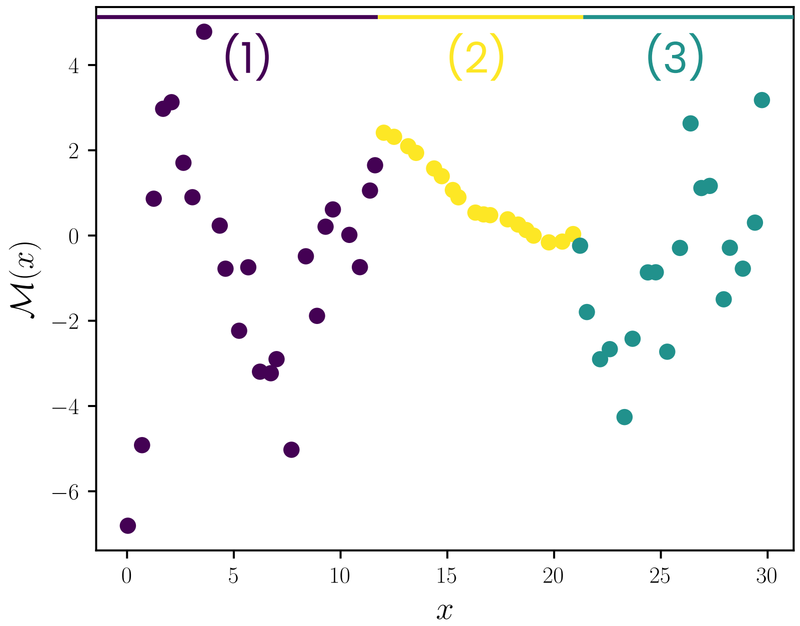
Feature space



Clustering step

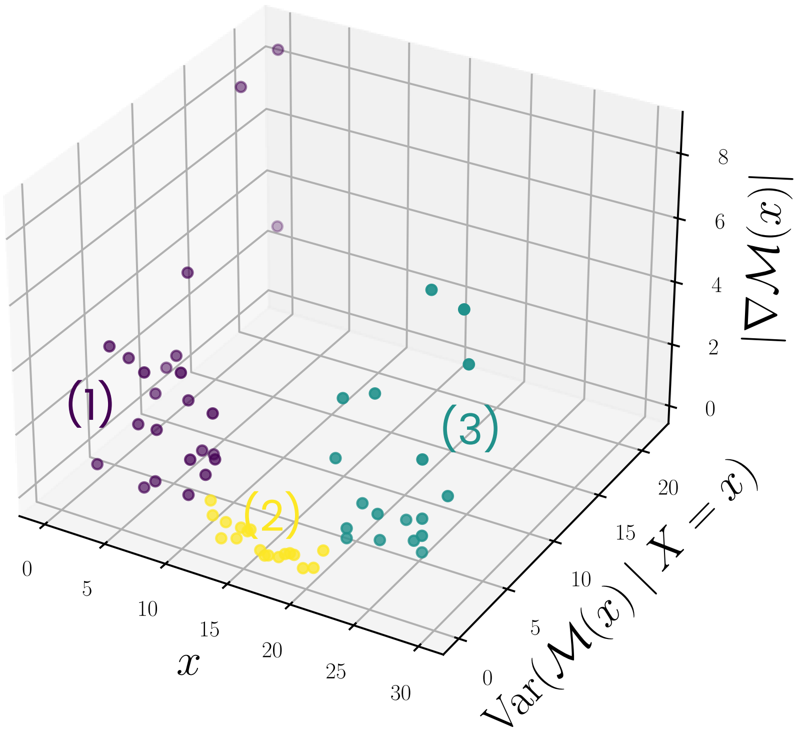
Clustering to detect stationary subregions

Observed data



Recover clusters in original space

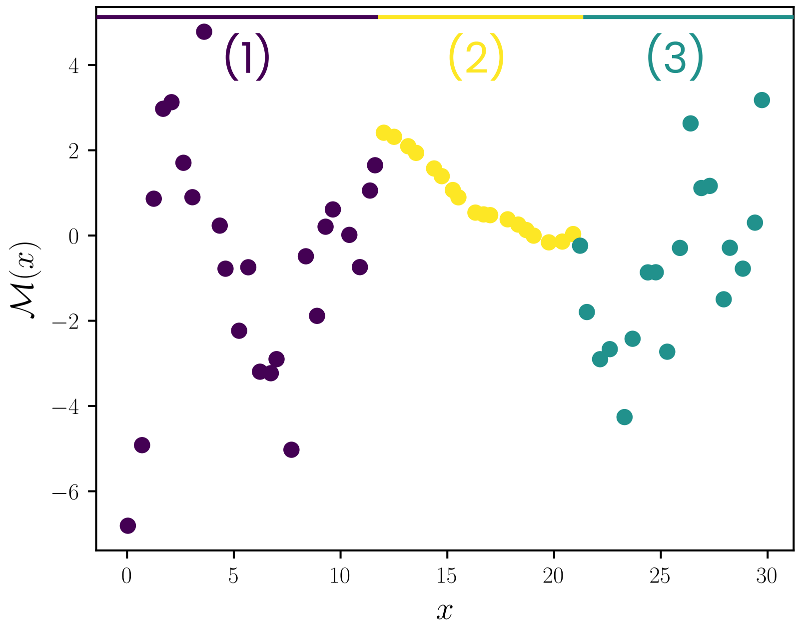
Feature space



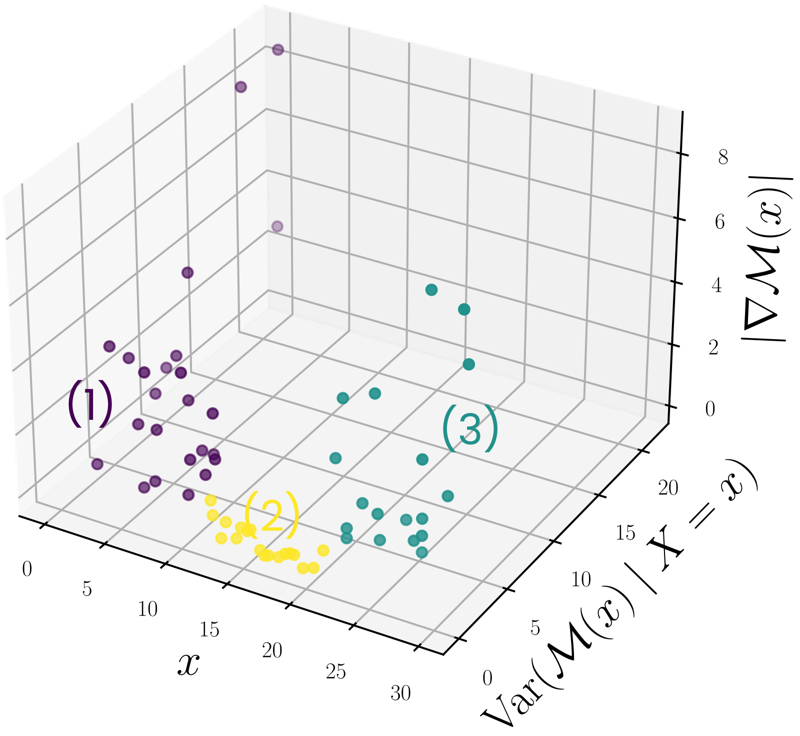
Clustering step

Clustering to detect stationary subregions

Observed data



Feature space



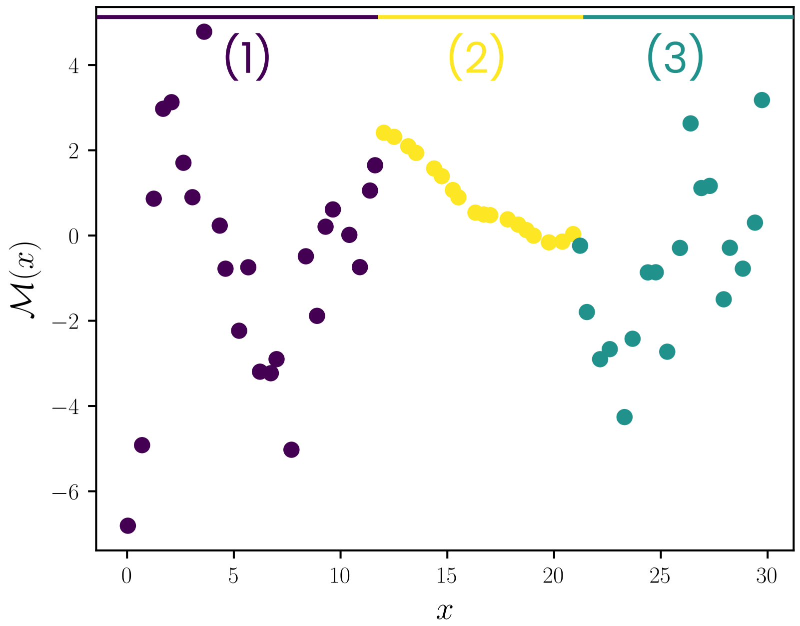
Recover clusters in original space

Clustering step

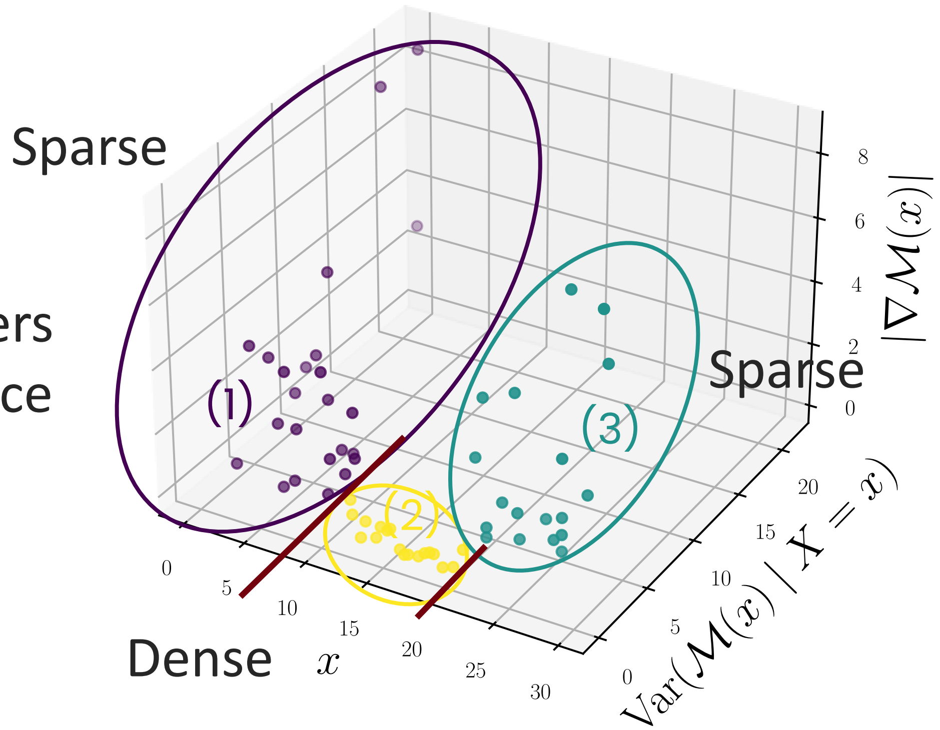
- Which clustering algorithm?

Clustering to detect stationary subregions

Observed data



Feature space



Recover clusters in original space

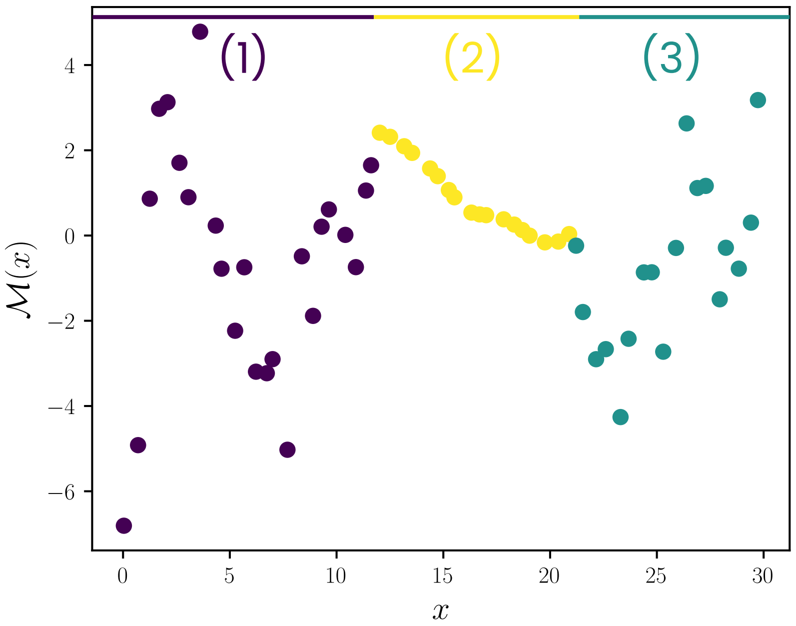
Clustering step

- Which clustering algorithm?

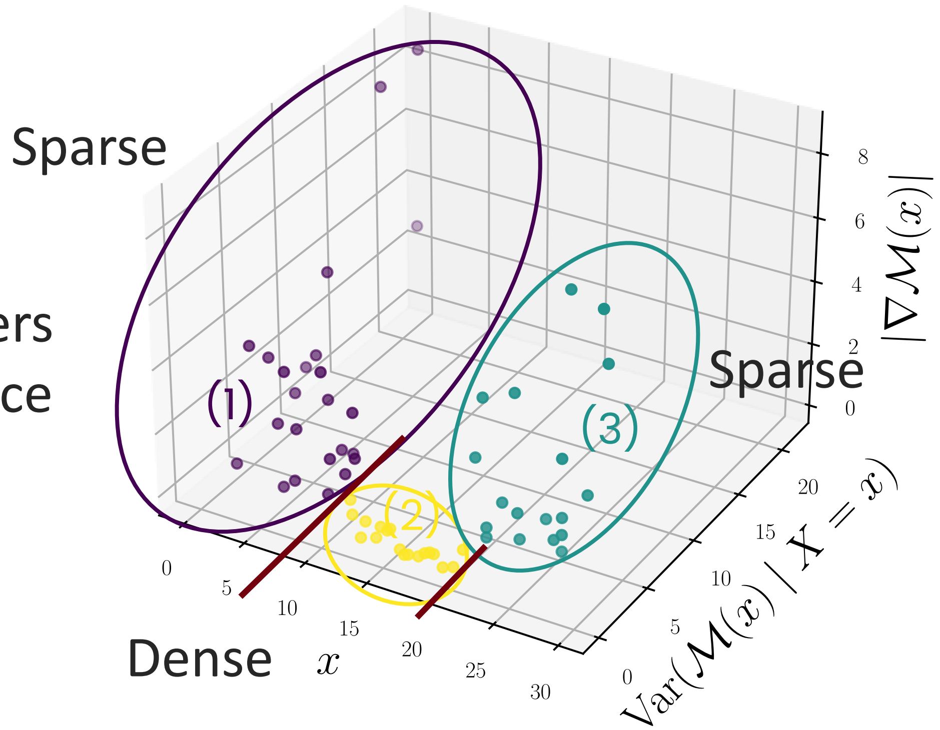
Prerequisites: Handling clusters with varying densities while enforcing spatial connectivity (L1)

Clustering to detect stationary subregions

Observed data



Feature space



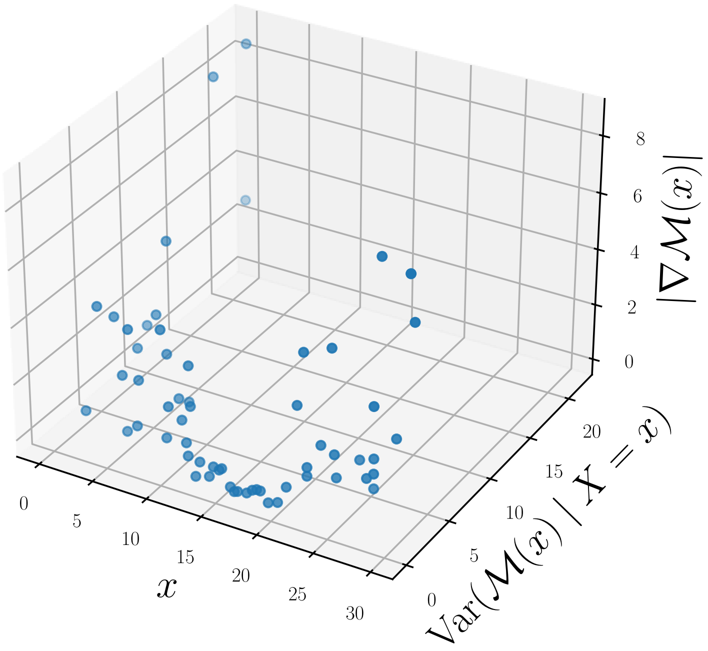
Recover clusters in original space

Clustering step

- Which clustering algorithm?
  - **Prerequisites:** Handling clusters with varying densities while enforcing spatial connectivity (L1)
    - **Proposed solution:** Spatially Constrained **Spectral clustering** using a connectivity graph → built-upon Voronoi tessellation

Spectral clustering algorithm in a nutshell

- Spectral clustering groups **similar** points together [vL07]

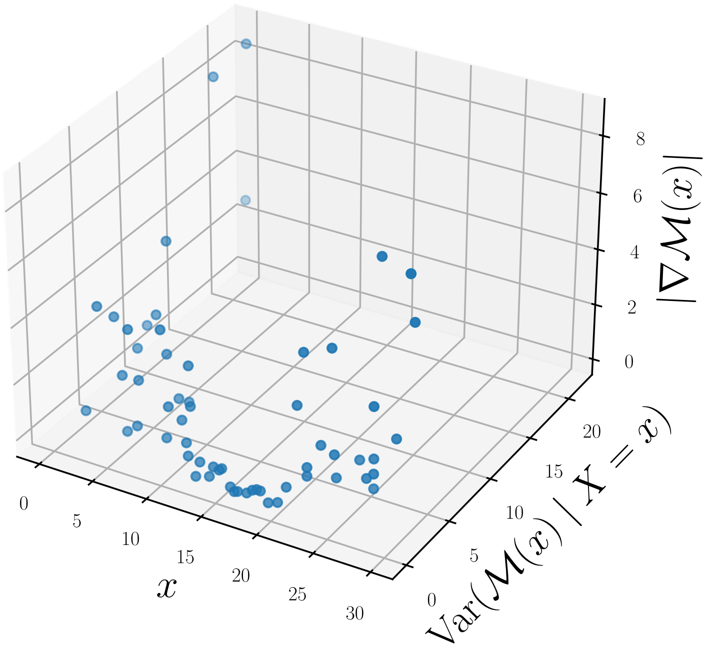


Features to cluster

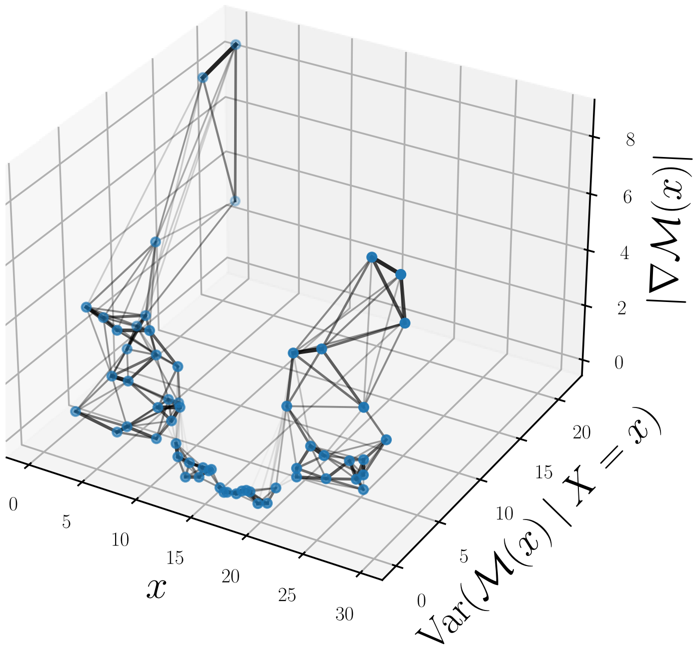
[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

Spectral clustering algorithm in a nutshell

- Spectral clustering groups **similar** points together [vL07]



Features to cluster

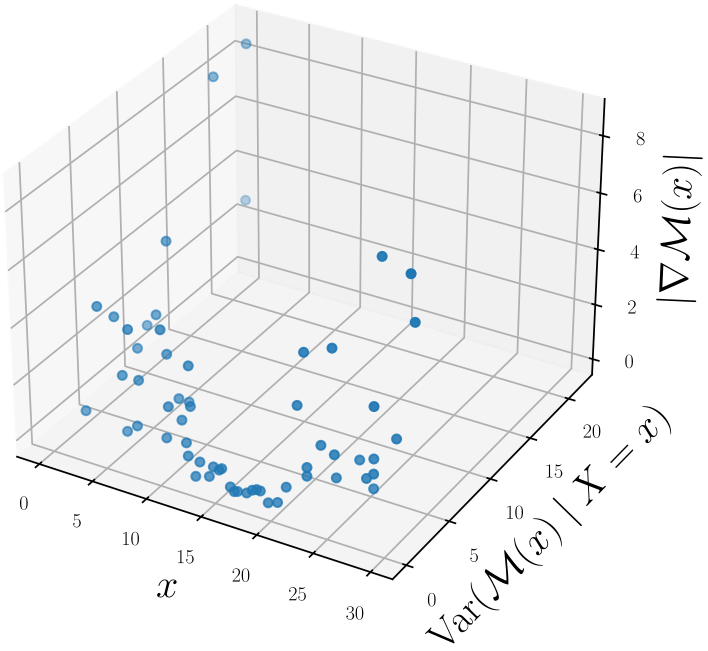


Compute pairwise similarity $s(i, j)$ between points $(\boldsymbol{z}_i, \boldsymbol{z}_j)$

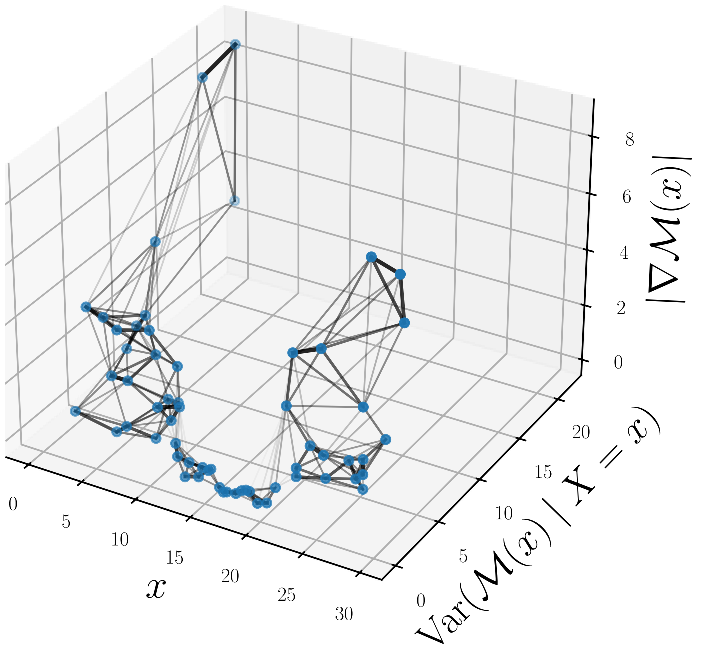
[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

Spectral clustering algorithm in a nutshell

- Spectral clustering groups **similar** points together [vL07]



Features to cluster

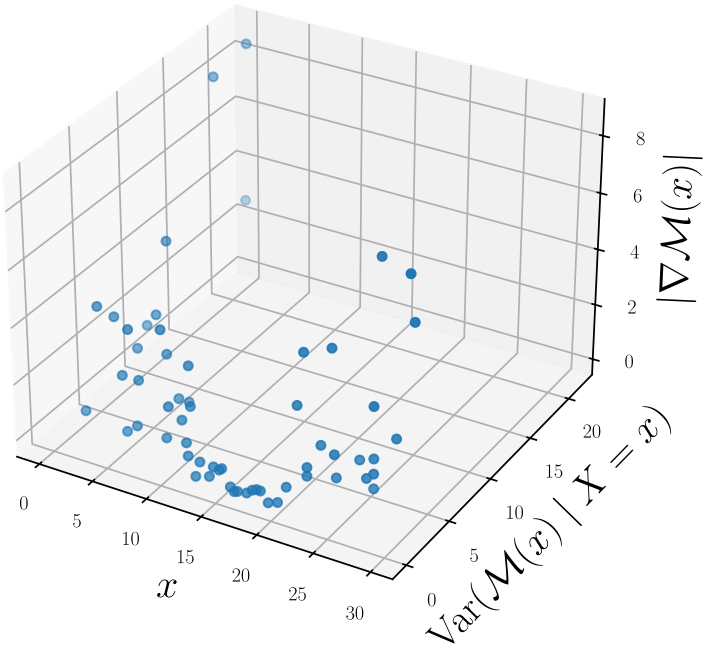


Compute pairwise similarity $s(i,j)$ between points $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ $\rightarrow$ Similarity matrix S + associated Laplacian L

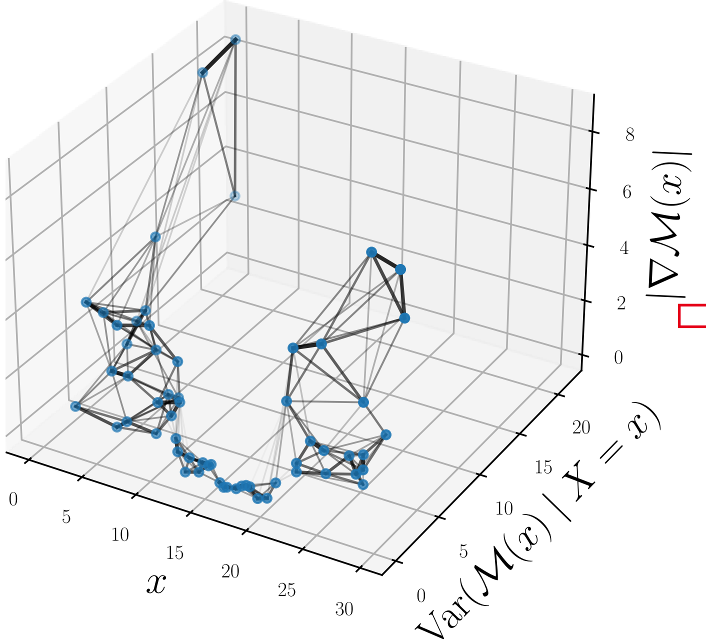
[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

Spectral clustering algorithm in a nutshell

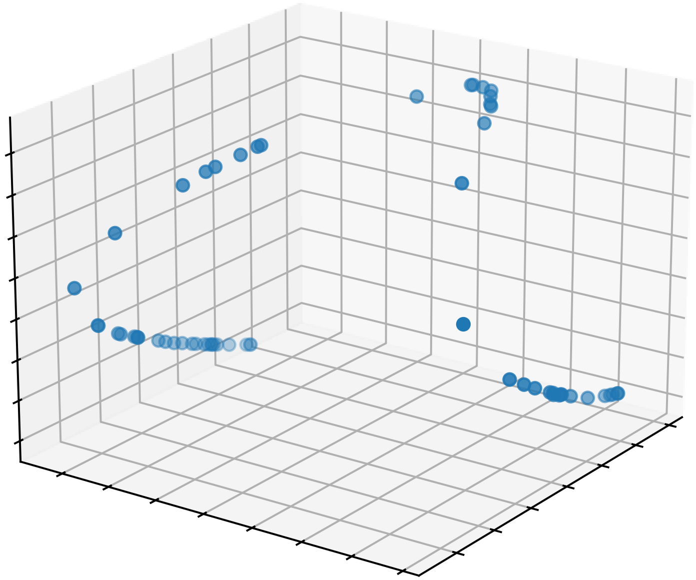
- Spectral clustering groups **similar** points together [vL07]



Features to cluster



Compute pairwise similarity $s(i, j)$ between points $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ → Similarity matrix S + associated Laplacian L



Spectral embedding from the Laplacian L eigenproblem :

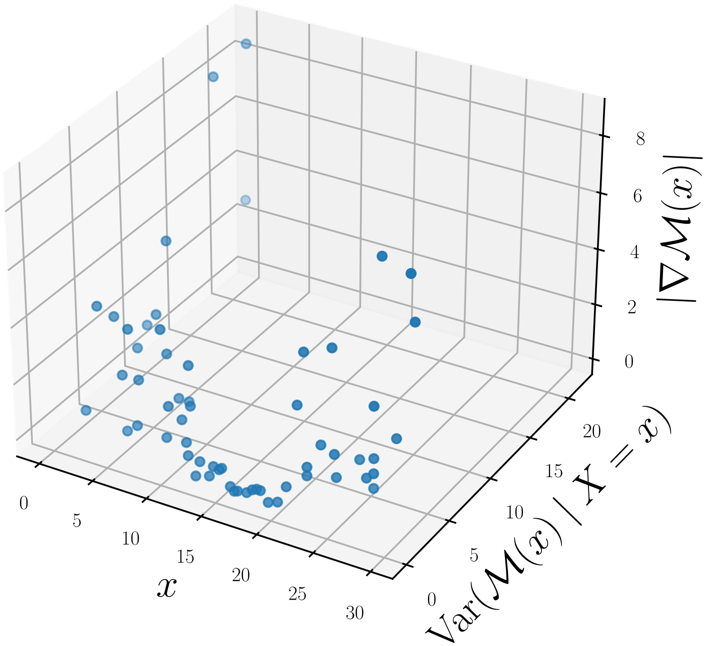
- First K eigenvectors of L
- Each point $\boldsymbol{z}_i \in \mathbb{R}^{2d+1}$ mapped to an embedding $\boldsymbol{e}_i \in \mathbb{R}^K$

[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

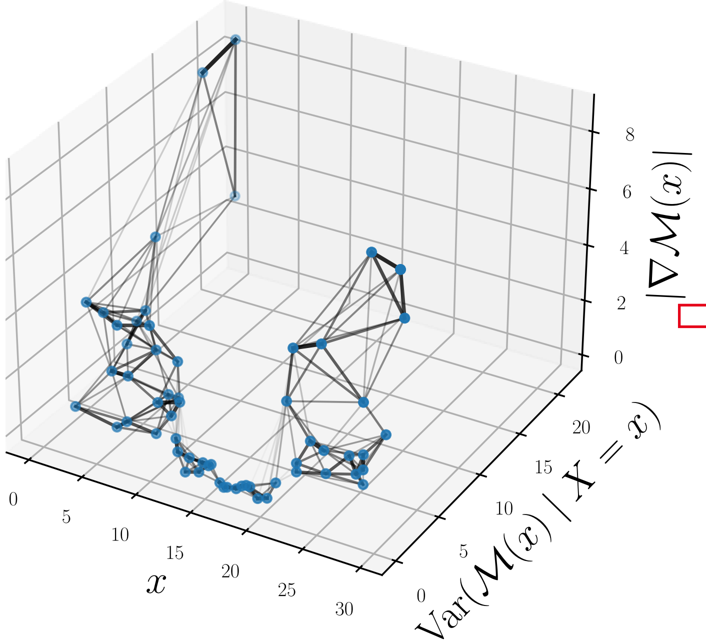
Spectral clustering algorithm in a nutshell

- Spectral clustering groups **similar** points together [vL07]

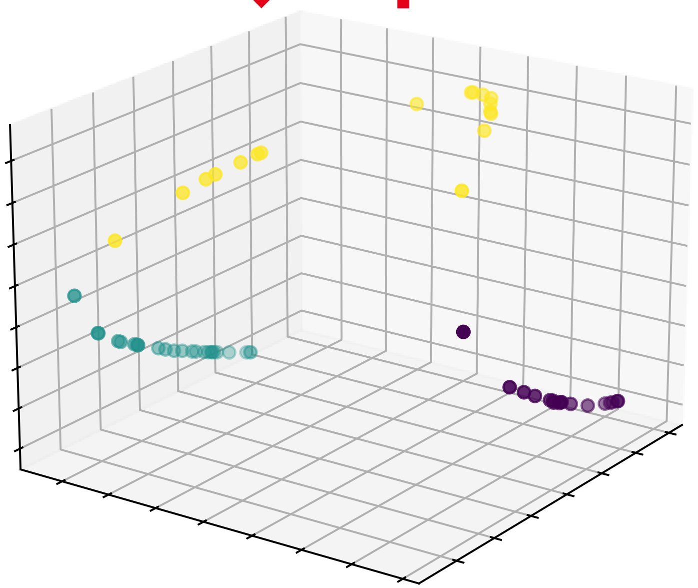
Spatially constrained (L1) hierarchical clustering to obtain clusters



Features to cluster



Compute pairwise similarity $s(i, j)$ between points $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ → Similarity matrix S + associated Laplacian L



Spectral embedding from the Laplacian L eigenproblem :

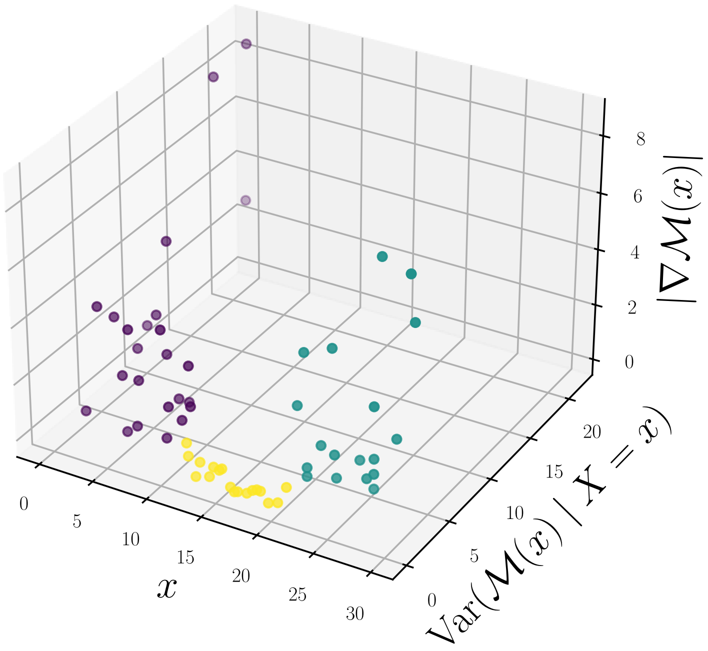
- First K eigenvectors of L
- Each point $\boldsymbol{z}_i \in \mathbb{R}^{2d+1}$ mapped to an embedding $\boldsymbol{e}_i \in \mathbb{R}^K$

[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

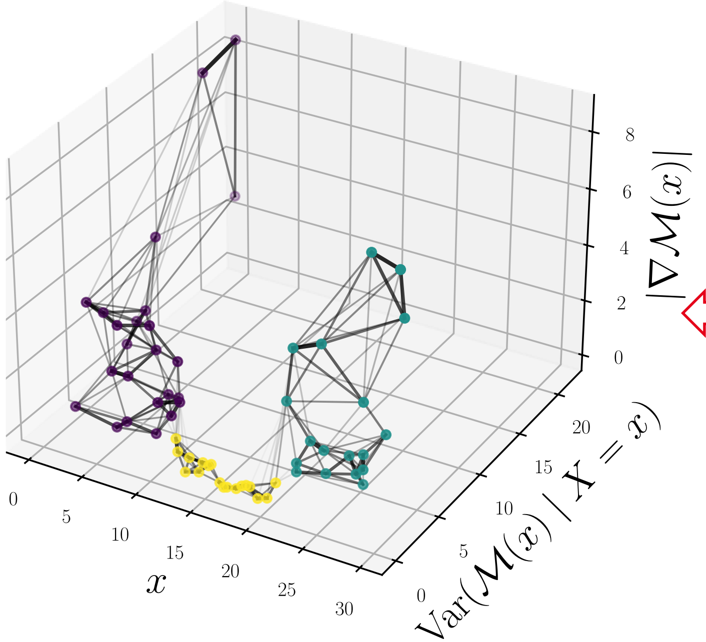
Spectral clustering algorithm in a nutshell

- Spectral clustering groups **similar** points together [vL07]

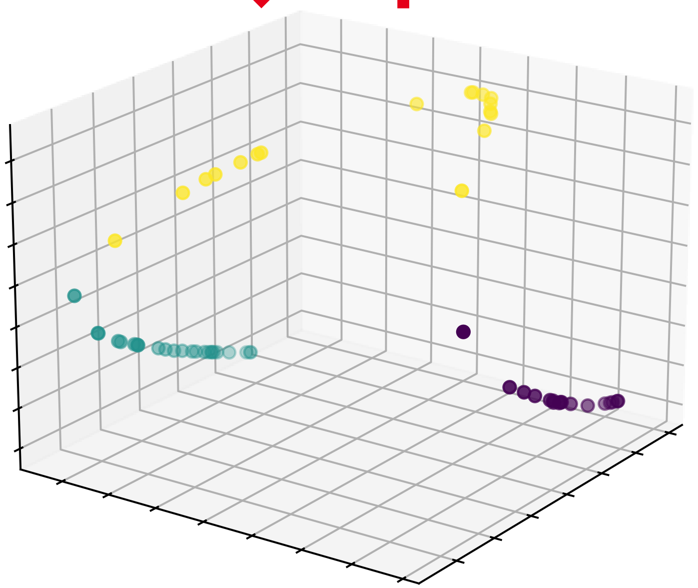
Spatially constrained (L1) hierarchical clustering to obtain clusters



Features to cluster



Compute pairwise similarity $s(i, j)$ between points $(\boldsymbol{z}_i, \boldsymbol{z}_j)$ → Similarity matrix S + associated Laplacian L



Spectral embedding from the Laplacian L eigenproblem :

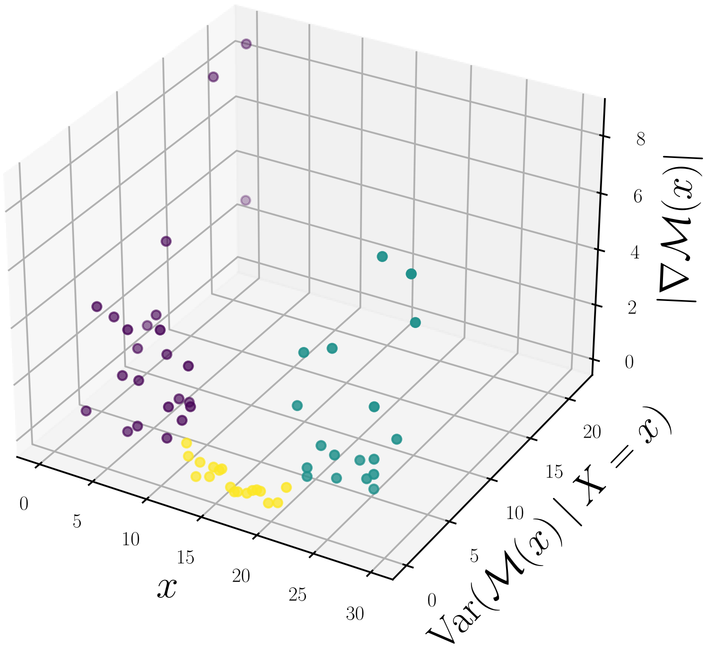
- First K eigenvectors of L
- Each point $\boldsymbol{z}_i \in \mathbb{R}^{2d+1}$ mapped to an embedding $\boldsymbol{e}_i \in \mathbb{R}^K$

[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

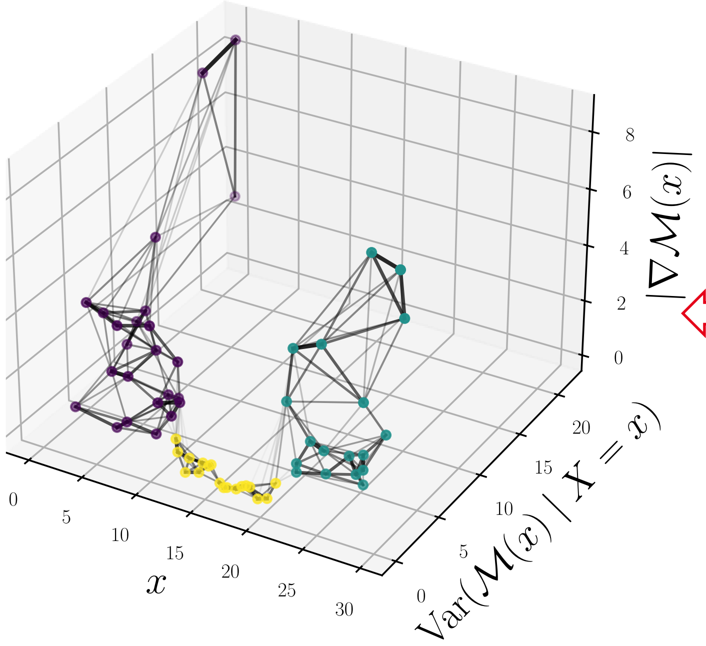
Spectral clustering algorithm in a nutshell

- Spectral clustering groups **similar** points together [vL07]

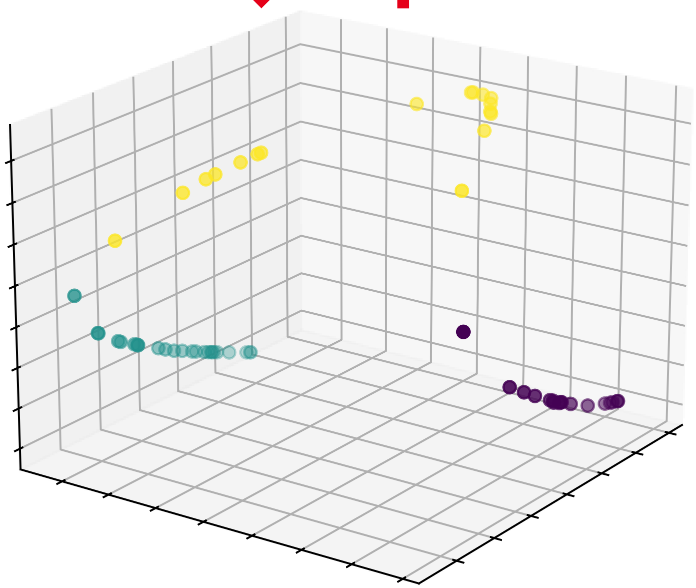
Spatially constrained (L1) hierarchical clustering to obtain clusters



Features to cluster



Compute pairwise similarity $s(i,j)$ between points $(\boldsymbol{z}_i, \boldsymbol{z}_j)$



Spectral embedding from the Laplacian L eigenproblem :

- First K eigenvectors of L
- Each point $\boldsymbol{z}_i \in \mathbb{R}^{2d+1}$ mapped to an embedding $\boldsymbol{e}_i \in \mathbb{R}^K$

- Which similarity function to use?

Combining spatial, variance and gradient similarities

- Similarity: $S(\boldsymbol{z}_i, \boldsymbol{z}_j) = S^{(\boldsymbol{x})}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, S^{(Var)}(v_i, v_j)\, S^{(\boldsymbol{\nabla})}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$

Combining spatial, variance and gradient similarities

- Similarity: $S(\boldsymbol{z}_i, \boldsymbol{z}_j) = S^{(x)}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, S^{(Var)}(v_i, v_j)\, S^{(\nabla)}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$

- Global input similarity $S^{(x)}(\boldsymbol{x}_i, \boldsymbol{x}_j)$:
  - Defined by 2 parameters

Combining spatial, variance and gradient similarities

- Similarity: $S(\boldsymbol{z}_i, \boldsymbol{z}_j) = S^{(\boldsymbol{x})}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, S^{(Var)}(v_i, v_j)\, S^{(\boldsymbol{\nabla})}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$

- Global input similarity $S^{(\boldsymbol{x})}(\boldsymbol{x}_i, \boldsymbol{x}_j)$:
  - ➢ Defined by 2 parameters
- Local feature similarity $S^{(Var)}(v_i, v_j) / S^{(\boldsymbol{\nabla})}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$: [ZmP+04]
  - ➢ 1 parameter defining the locality per feature

[ZmP+04] Zelnik-manor, L., Perona, P. (2004). Self-Tuning Spectral Clustering. MIT Press

Combining spatial, variance and gradient similarities

- Similarity: $S(\boldsymbol{z}_i, \boldsymbol{z}_j) = S^{(x)}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, S^{(Var)}(v_i, v_j)\, S^{(\nabla)}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$

- Global input similarity $S^{(x)}(\boldsymbol{x}_i, \boldsymbol{x}_j)$:
  - Defined by 2 parameters
- Local feature similarity $S^{(Var)}(v_i, v_j)/\, S^{(\nabla)}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$: [ZmP+04]
  - 1 parameter defining the locality per feature

- In total, 4 similarity parameters must be fine-tuned:
  - For different number of clusters, tuned following [FTZ+22]

[ZmP+04] Zelnik-manor, L., Perona, P. (2004). Self-Tuning Spectral Clustering. MIT Press
[FTZ+22] Fan J., Tu, Y., Zhang, Z., Zhao, M., Zhang, H. (2022). A Simple Approach to Automated Spectral Clustering. Curran Associates, Inc.

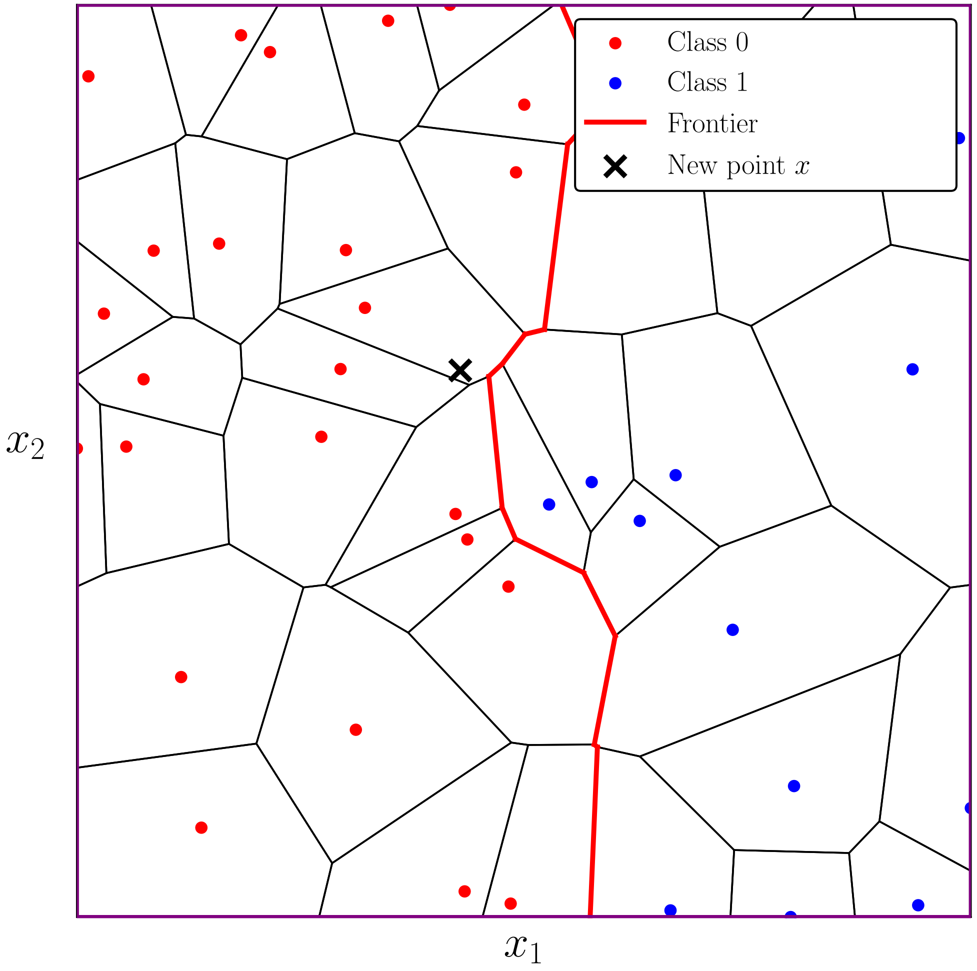
Step 2: Classification

The Voronoi-based classifier

- The classifier assigns each new input $\boldsymbol{x}$ to a stationary subregion for prediction

- The classifier decision boundary (frontier) is given by the union of the Voronoi cell boundaries separating different classes

- Membership probabilities derived from distances to decision boundaries [P99] [WLX04]



2D Voronoi-based classifier

[P99] Platt, J. C., (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In Advances in Large Margin Classifiers, MIT Press.
[WLX04] Wu, T., Lin, C., Weng, C. (2004). Probability estimates for multi-class classification by pairwise coupling. JMLR.

Step 3: Final non-stationary $\mathcal{GP}$ surrogate

Gaussian Process Mixture Model

- Membership probabilities for cluster C_k:

$$\pi_k(\boldsymbol{x}) = P(\boldsymbol{x} \in C_k),\ k = 1, \dots, K$$

- Mixture predictive distribution at $\boldsymbol{x}$:

$$p(.|\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\ \mathcal{N}\left(.\,; \mu_k(\boldsymbol{x}), \sigma_k^2(\boldsymbol{x})\right)$$

  - Local predictor of C_k: $\mu_k(\boldsymbol{x})$
  - Local predictive variance of C_k: $\sigma_k^2(\boldsymbol{x})$

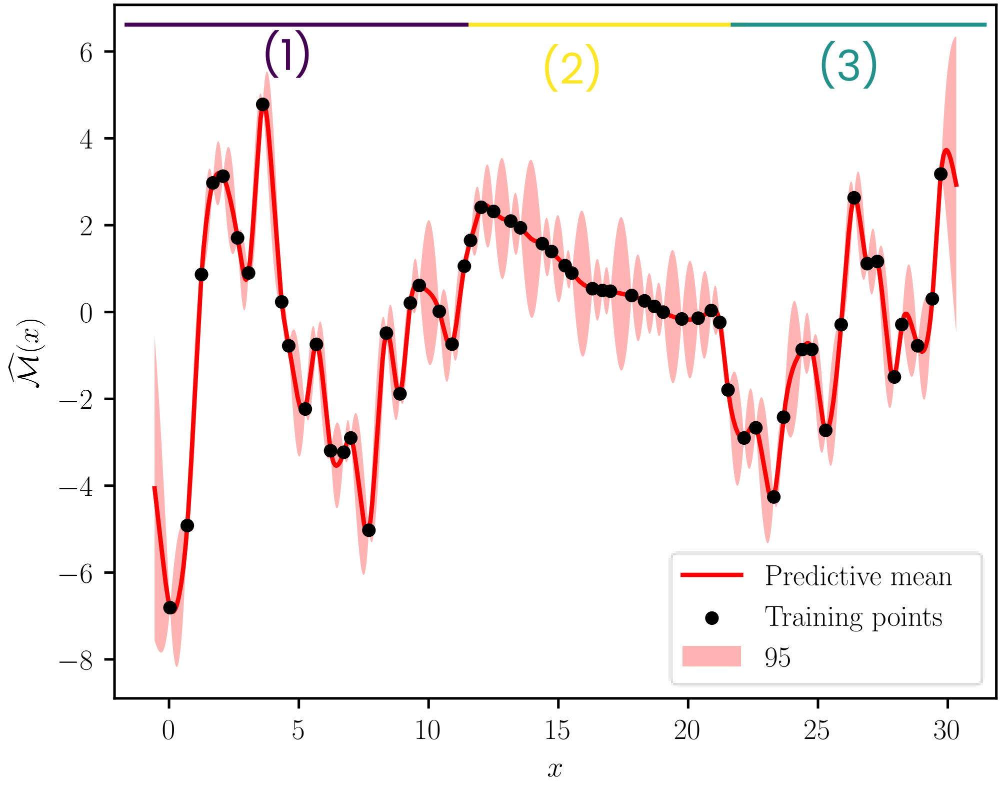
- Mixture predictor:

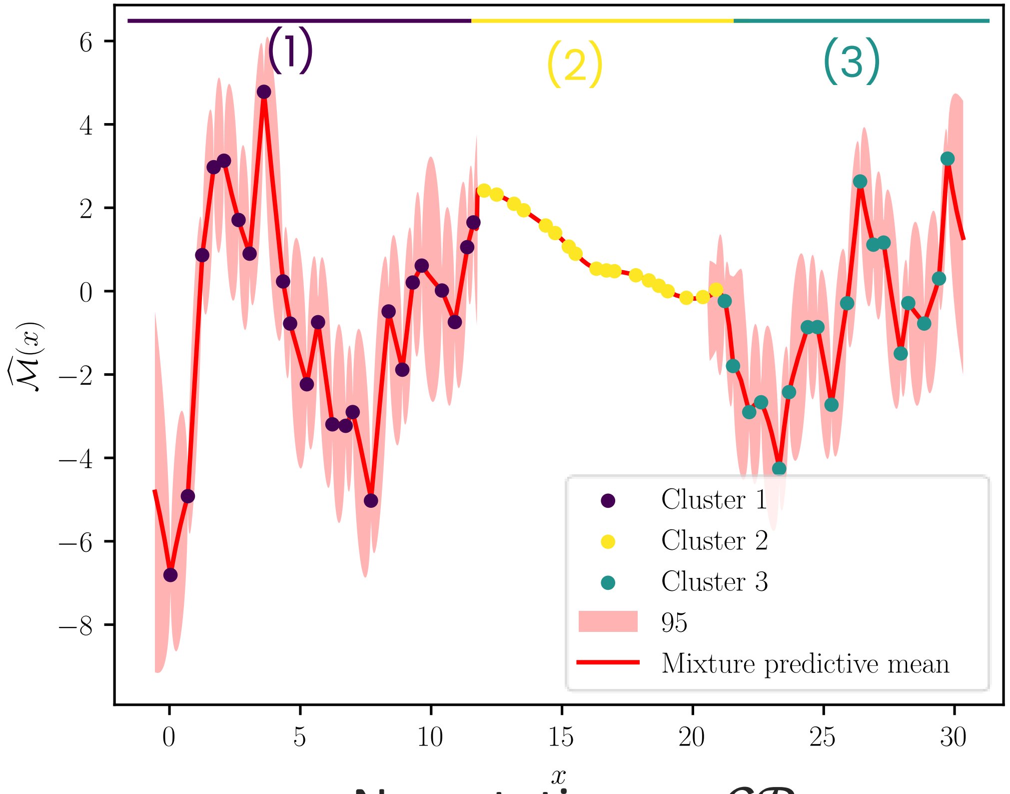
$$\mu_{mix}(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\,\mu_k(\boldsymbol{x})$$

- Mixture predictive variance:

$$\sigma_{mix}^2(\boldsymbol{x}) = \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\sigma_k^2(\boldsymbol{x}) + \sum_{k=1}^{K} \pi_k(\boldsymbol{x})\big(\mu_k(\boldsymbol{x}) - \mu_{mix}(\boldsymbol{x})\big)^2$$

Regression results on the 1D non-stationary function





Single stationary $\mathcal{GP}$

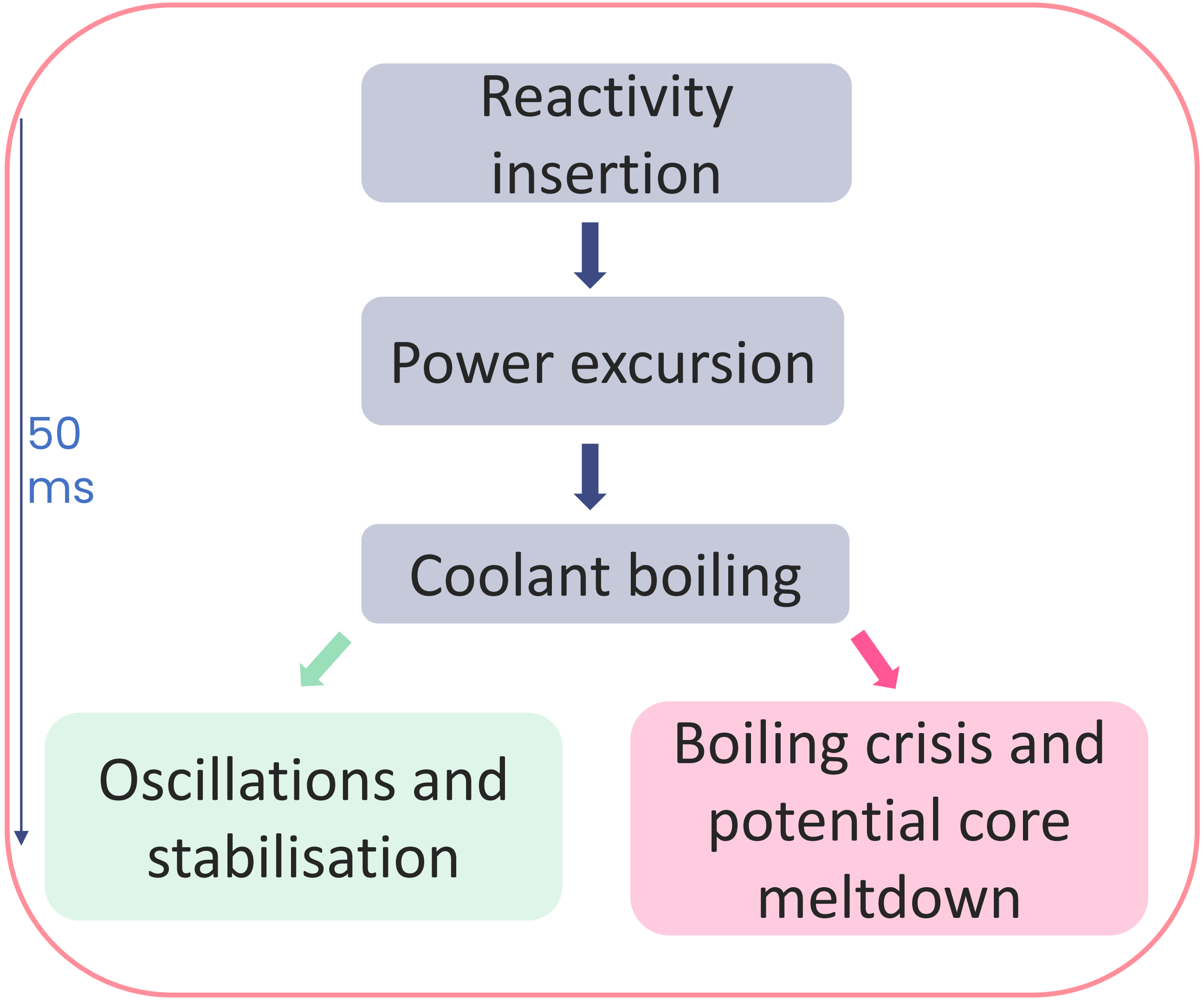
- Zone (2): Under confident ✗
- Zones (1), (3): Over confident ✗

Non-stationary $\mathcal{GP}$

- Zone (2): More confident ✓
- Zones (1), (3): Less confident ✓

Application to a nuclear accident

Application to a nuclear accident



- Non-stationarity is due to rapid power excursion and changes in heat exchanges

Regression problem parameters :

- 3D inputs
- Learning size: 275
- Validation size: 45
- Tested number of clusters: $K \in [\![2,5]\!]$

Surrogate modeling results

- Standard validation metrics for probabilistic surrogates: [DIG+22]

➢ $Q^2 = 1 - \dfrac{\mathbb{E}\left[\left(\mathcal{M}(\mathbf{x}) - \mathbb{E}[\mathcal{M}(\mathbf{x}) \mid \mathbf{x}]\right)^2\right]}{\mathrm{Var}(\mathcal{M}(\mathbf{x}))} \in]-\infty, 1]$

➢ $\mathrm{IAE} = \int_0^1 \left| C(\alpha) - \alpha \right| \, \mathrm{d}\alpha \in [0, 0.5]$

Coverage of predictive intervals

$\mathcal{GP}$ Surrogate		Q^2	IAE
Stationary	Standard $\mathcal{GP}$	0.79	0.10
Non-stationary	Spectral clustering Mix $\mathcal{GP}$	**0.84**	0.08
	Treed $\mathcal{GP}$ [GL08]	0.80	**0.04**

[DIG+22] Demay, C., Iooss, B., Gratiet, L., and Marrel, A. (2022). Model selection for GP regression: an application with highlights on the model variance validation. QREI Journal, 38:1482-1500.
[GL08] Gramacy, Robert B., Lee, Herbert K. H. (2008). Bayesian Treed Gaussian Process Models with an Application to Computer Modeling. Journal of the American Statistical Association.

Conclusions and perspective

Conclusions:

- Improvement of the three-step approach of [MS23] for non-stationary $\mathcal{GP}$ modeling
- The clustering step now combines stationarity-related local features with spatially constrained spectral clustering, ensuring path-connected stationary sub-regions
- On the real dataset, the proposed non-stationary model:
  - Improved predictive accuracy and variance calibration compared with a stationary $\mathcal{GP}$ model
  - Showed better predictive performance than Treed $\mathcal{GP}$ model

Perspective:

- Improve predictive performance by adding points to clusters that are the most difficult to predict

[MS23] Moustapha, M., Sudret, B. (2023). Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling. Computers & Structures, Elsevier.

Thank you for your attention

Any questions?

CEA SACLAY
91191 Gif-sur-Yvette Cedex
France
theo.sylvestre@cea.fr

References

[DIG+22] Demay, C., Iooss, B., Gratiet, L., and Marrel, A. (2022). Model selection for GP regression: an application with highlights on the model variance validation. QREI Journal, 38:1482-1500.

[FTZ+22] Fan J., Tu, Y., Zhang, Z., Zhao, M., Zhang, H. (2022). A Simple Approach to Automated Spectral Clustering. Curran Associates, Inc.

[GL08] Gramacy, Robert B., Lee, Herbert K. H. (2008). Bayesian Treed Gaussian Process Models with an Application to Computer Modeling. Journal of the American Statistical Association.

[MS23] Moustapha, M., Sudret, B. (2023). Learning non-stationary and discontinuous functions using clustering, classification and Gaussian process modelling. Computers & Structures, Elsevier.

[P99] Platt, J. C., (1999). Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. In Advances in Large Margin Classifiers, MIT Press.

[RW06] Rasmussen, C.E., Williams, C.K.I. (2006). Gaussian processes for machine learning. MIT Press.

[vL07] von Luxburg, U. (2007). A tutorial on spectral clustering. Springer.

[WLX04] Wu, T., Lin, C., Weng, C. (2004). Probability estimates for multi-class classification by pairwise coupling. JMLR

[ZmP+04] Zelnik-manor, L., Perona, P. (2004). Self-Tuning Spectral Clustering. MIT Press

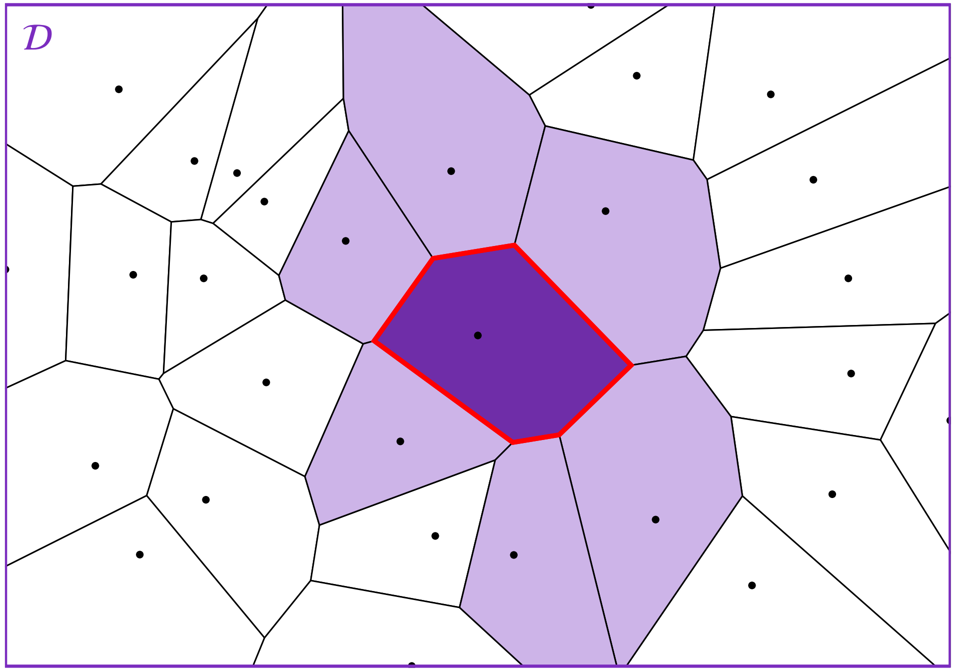
The constrained Voronoi tessellation

- Admissible space: $\mathcal{D} = \prod_{j=1}^{d} \left[\min_{1 \leq i \leq n} x_{ij}, \ \max_{1 \leq i \leq n} x_{ij} \right].$

- The Voronoi cell of x_i:

$$V_i^D = \{x \in D \ : \ \|x - x_i\|_2 \leq \|x - x_j\|_2, \quad \forall j \neq i\}$$

- Connectivity matrix derived from the set of neighbors of each point



2D Voronoi tessellation

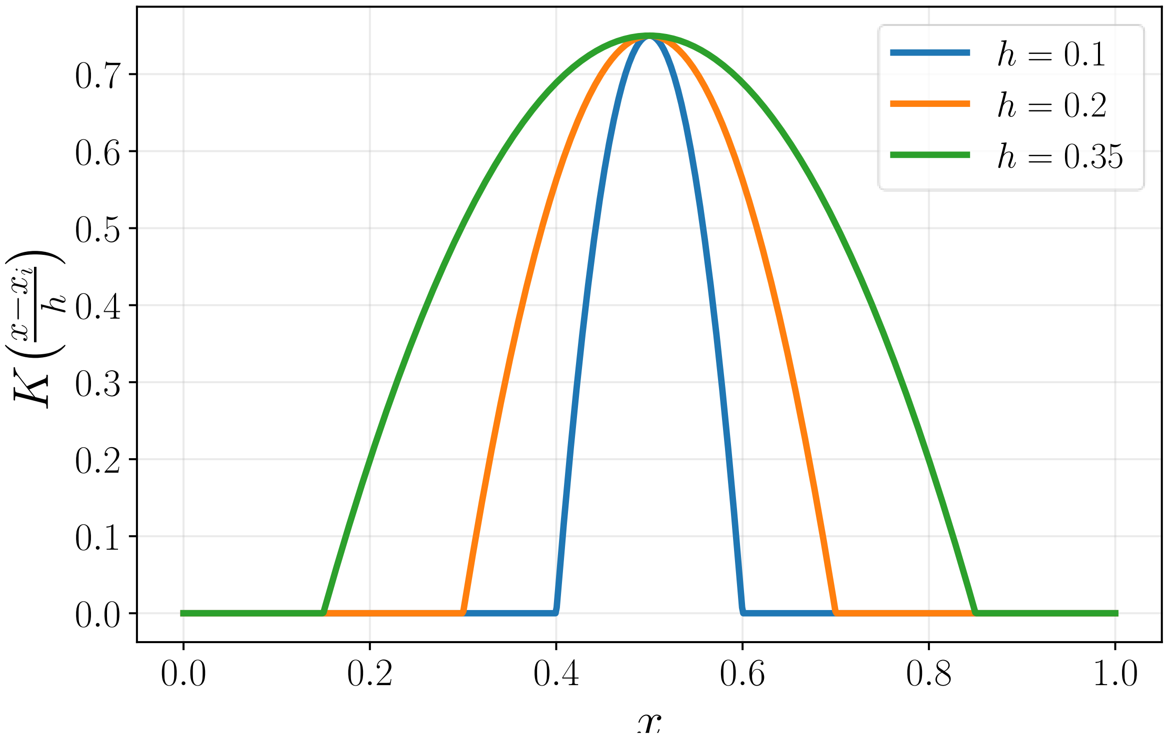
Local variance estimation

- For $\boldsymbol{x}_i \in \mathbb{R}^d, h_i^{(k)} = \left\| \boldsymbol{x}_i - \boldsymbol{x}_i^{(k)} \right\|, \boldsymbol{x}_i^{(k)} = k^{th}$ nearest neighbors of $\boldsymbol{x}_i$
- Let $w_{i,j}^{(k)} = K\left(\frac{\|\boldsymbol{x}_j - \boldsymbol{x}_i\|}{h_i^{(k)}}\right)$
- $\forall\, u \in \mathbb{R}, K(u) = \frac{3}{4}(1-u^2)1_{(|u|\leq 1)}$
  - ➢ Epanechnikov Kernel
- Kernel smoothing estimator of the mean:

$$\widehat{m}_{-i}^{(k)}(\boldsymbol{x}_i) = \sum_{j \neq i} \frac{w_{i,j}^{(k)} y_j}{\sum_{j \neq i} w_{i,j}^{(k)}}$$

- Kernel smoothing estimator of the variance:

$$\widehat{\mathrm{Var}}_{-i}^{(k)}(\boldsymbol{x}_i) = \frac{\sum_{j \neq i} w_{i,j}^{(k)} \left(y_j - \widehat{m}_{-j}^{(k)}(\boldsymbol{x}_j)\right)^2}{\sum_{j \neq i} w_{i,j}^{(k)}}$$



Epanechnikov kernel with varying bandwidth h

Spectral clustering as a minimisation problem

- Goal: Partition similar points into $C_1, \dots, C_K$ clusters

  ↳ Similarity function

- Similarity matrix $S = \left(\mathbf{1}_{i \neq j} s_{i,j}\right)_{i,j \in [\![1,n]\!]^2}$

- Degree of $\mathbf{z}_\mathrm{i}$: $d_i = \sum_{j=1}^{n} s_{ij}$
  - Degree matrix $\mathrm{D} = Diag(d_1, \dots, d_n)$

- Laplacian matrix: $L = I_n - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$

- $\mathrm{H} = \left(\frac{\sqrt{d_i}}{\sqrt{\sum_{i \in C_k} d_i}} \mathbf{1}_{v_i \in C_k} \right)_{i,k \in [\![1,n]\!] \times [\![1,K]\!]}$

Spectral clustering as a minimisation problem

- Goal: Partition similar points into $C_1, \dots, C_K$ clusters

  ⮡ Similarity function

- Similarity matrix $S = \left(\mathbf{1}_{i \neq j} s_{i,j}\right)_{i,j \in [\![1,n]\!]^2}$
- Degree of $\mathbf{z}_\mathrm{i}$: $d_i = \sum_{j=1}^{n} s_{ij}$
  - Degree matrix $\mathrm{D} = Diag(d_1, \dots, d_n)$
- Laplacian matrix: $L = I_n - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$
- $\mathrm{H} = \left(\frac{\sqrt{d_i}}{\sqrt{\sum_{i \in C_k} d_i}} \mathbf{1}_{v_i \in C_k}\right)_{i,k \in [\![1,n]\!] \times [\![1,K]\!]}$

Discret spectral clustering problem

$$\boxed{\mathrm{C}_1, \dots, \mathrm{C}_\mathrm{K} = \underset{C_1, \dots, C_k}{\operatorname{argmin}} \, Tr(H^T L H) \; s.t \; H^T H = I}$$

Spectral clustering as a minimisation problem

- Goal: Partition similar points into $C_1, \dots, C_K$ clusters

  Similarity function

- Similarity matrix $S = \left(\mathbf{1}_{i \neq j} s_{i,j}\right)_{i,j \in [\![1,n]\!]^2}$

- Degree of $\mathbf{z}_i$: $d_i = \sum_{j=1}^{n} s_{ij}$
  - Degree matrix $D = Diag(d_1, \dots, d_n)$

- Laplacian matrix: $L = I_n - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$

- $H = \left(\frac{\sqrt{d_i}}{\sqrt{\sum_{i \in C_k} d_i}} \mathbf{1}_{v_i \in C_k} \right)_{i,k \in [\![1,n]\!] \times [\![1,K]\!]}$

Discret spectral clustering problem

$$\boxed{C_1, \dots, C_K = \underset{C_1, \dots, C_k}{\operatorname{argmin}}\, Tr(H^T L H) \ s.t \ H^T H = I}$$

NP hard problem! → Relaxed optimisation problem

Spectral clustering as a minimisation problem

- Goal: Partition similar points into $C_1, \dots, C_K$ clusters

  ↳ Similarity function

- Similarity matrix $S = \left(\mathbf{1}_{i\neq j} s_{i,j}\right)_{i,j\in[\![1,n]\!]^2}$

- Degree of $\mathbf{z}_\mathrm{i}$: $d_i = \sum_{j=1}^{n} s_{ij}$
  - Degree matrix $\mathrm{D} = Diag(d_1, \dots, d_n)$

- Laplacian matrix: $L = I_n - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$

- $\mathrm{H} = \left(\frac{\sqrt{d_i}}{\sqrt{\sum_{i\in C_k} d_i}} \mathbf{1}_{v_i\in C_k}\right)_{i,k\in[\![1,n]\!]\times[\![1,K]\!]}$

Continious spectral clustering problem

$$\boxed{\mathrm{C}_1, \dots, \mathrm{C}_\mathrm{K} = \underset{H\in\mathcal{M}_{n\times K}(\mathbb{R})}{\operatorname{argmin}}\, Tr(H^T L H)\ s.t\ H^T H = I}$$

Spectral clustering as a minimisation problem

- Goal: Partition similar points into $C_1, \ldots, C_K$ clusters

  ↳ Similarity function

- Similarity matrix $S = \left(\mathbf{1}_{i \neq j} s_{i,j}\right)_{i,j \in [\![1,n]\!]^2}$

- Degree of $\mathbf{z}_\mathrm{i}$: $d_i = \sum_{j=1}^{n} s_{ij}$
  - Degree matrix $\mathrm{D} = Diag(d_1, \ldots, d_n)$

- Laplacian matrix: $L = I_n - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$

- $\mathrm{H} = \left(\frac{\sqrt{d_i}}{\sqrt{\sum_{i \in C_k} d_i}} \mathbf{1}_{v_i \in C_k}\right)_{i,k \in [\![1,n]\!] \times [\![1,K]\!]}$

Continious spectral clustering problem

$$\mathrm{C}_1, \ldots, \mathrm{C}_\mathrm{K} = \underset{H \in \mathcal{M}_{n \times K}(\mathbb{R})}{\operatorname{argmin}} Tr(H^T L H) \ s.t \ H^T H = I$$

- Relaxed solution: K first eigenvectors $e_1, \ldots e_K$ of L

Spectral clustering as a minimisation problem

- Goal: Partition similar points into $C_1, \dots, C_K$ clusters

  ↳ Similarity function

- Similarity matrix $S = \left(\mathbf{1}_{i \neq j} s_{i,j}\right)_{i,j \in [\![1,n]\!]^2}$

- Degree of $\mathbf{z}_i$: $d_i = \sum_{j=1}^{n} s_{ij}$
  - Degree matrix $D = Diag(d_1, \dots, d_n)$

- Laplacian matrix: $L = I_n - D^{-\frac{1}{2}} S D^{-\frac{1}{2}}$

- $H = \left(\frac{\sqrt{d_i}}{\sqrt{\sum_{i \in C_k} d_i}} \mathbf{1}_{v_i \in C_k} \right)_{i,k \in [\![1,n]\!] \times [\![1,K]\!]}$

Continious spectral clustering problem

$$\boxed{C_1, \dots, C_K = \underset{H \in \mathcal{M}_{n \times K}(\mathbb{R})}{\operatorname{argmin}} Tr(H^T L H) \; s.t \; H^T H = I}$$

- Relaxed solution: K first eigenvectors $e_1, \dots e_K$ of L

- Discrete solution via K-cluster partitioning of $e_1, \dots e_K$

Tuning similarity parameters

- For a fixed number of cluster K:

$$l, \sigma_X, c_v\,, c_{\nabla} = \underset{l,\sigma_X,c_v\,,\,c_{\nabla}}{\operatorname{argmax}}\ \lambda_{k+1} - \frac{1}{K}\sum_{k=1}^{K} \lambda_k$$

Least connectivity among clusters

Connectivity among clusters

- $\lambda_1 \leq \ \ldots\ldots \leq \lambda_n$ = eigenvalues of the Laplacian matrix L derived from similarities

Choosing the number of clusters

- For $\boldsymbol{x}_i \in \mathbb{R}^d$
- Within-cluster density distance:

$$a_i = \frac{1}{\left|C_{cluster(\boldsymbol{x}_i)}\right| - 1} \sum_{x_j \in C_{cluster(x_i)}, x_j \neq x_i} d_{density}(\boldsymbol{x}_j, \boldsymbol{x}_i)$$

→ Measures a point's within-cluster density based distance

- Nearest-cluster density distance:

$$b_i = min_{C \neq C_{cluster(x_i)}} \left(\frac{1}{|C|} \sum_{x_j \in C} d_{density}(\boldsymbol{x}_i, \boldsymbol{x}_j) \right)$$

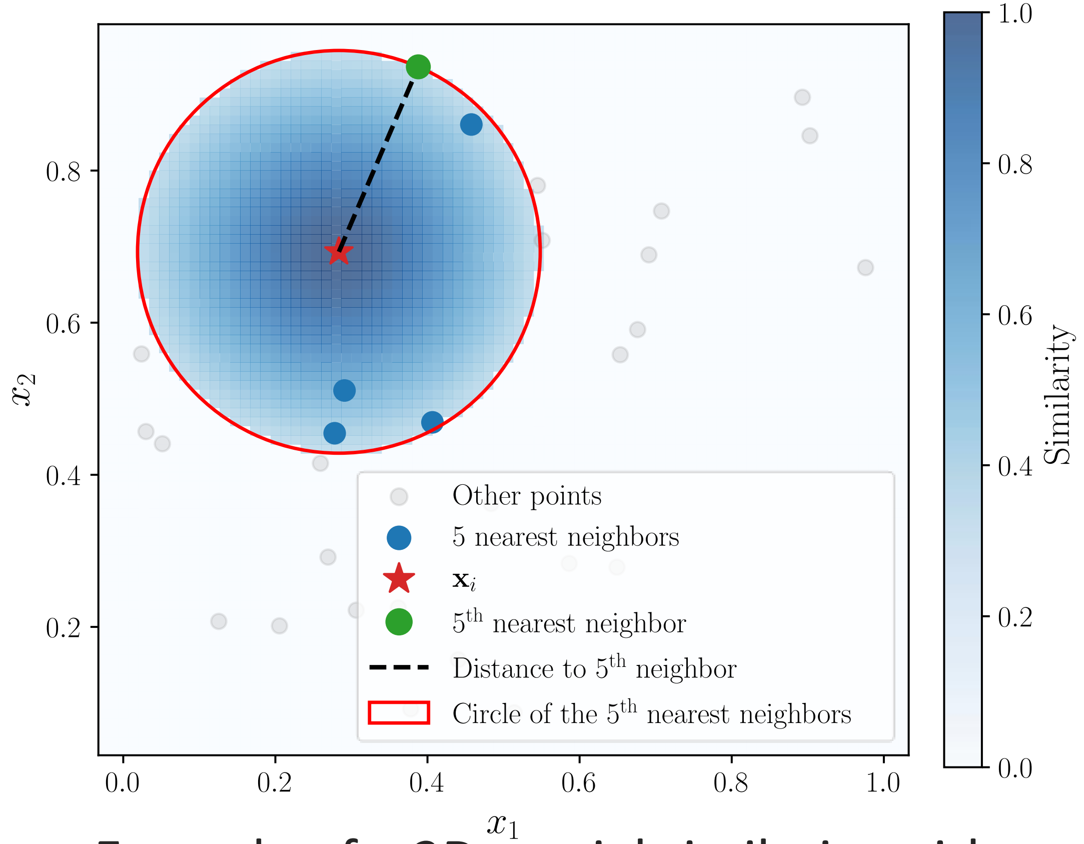
→ Measures a point's density based distance from other clusters

- Final DBCV index:

$$DBCV = \frac{1}{n} \sum_{i=1}^{n} \frac{b_i - a_i}{max(a_i, b_i)} \in [-1,1]$$

→ The closer to 1 the better

Combining spatial, variance and gradient similarities



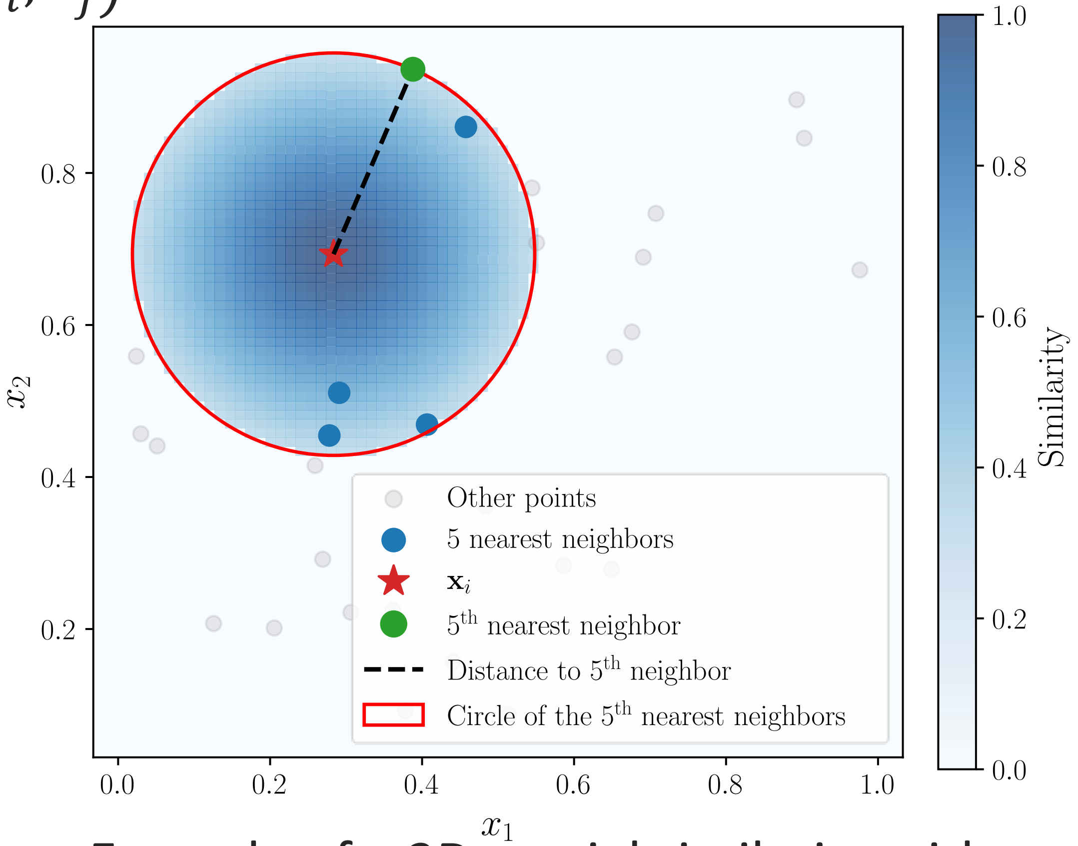
Example of a 2D spatial similarity with $l = 5, \sigma_X = 0.25$

Combining spatial, variance and gradient similarities

- Similarity: $S(\boldsymbol{z}_i, \boldsymbol{z}_j) = S^{(x)}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, S^{(Var)}(v_i, v_j)\, S^{(\nabla)}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$

$$\mathbb{1}\left(\boldsymbol{x}_j \in \mathcal{N}_{l(\boldsymbol{x}_i)} \text{ or } \boldsymbol{x}_i \in \mathcal{N}_{l(\boldsymbol{x}_j)}\right) \exp\left(-\frac{\left\|\boldsymbol{x}_i - \boldsymbol{x}_j\right\|_2^2}{\sigma_X^2}\right)$$

  - $\mathcal{N}_l(\boldsymbol{x}_i) = \{\text{the } l \text{ nearest neighbors of } \boldsymbol{x}_i\}$
  - σ_x, l must be tuned



Example of a 2D spatial similarity with $l = 5, \sigma_X = 0.25$

Combining spatial, variance and gradient similarities

- Similarity: $S(\boldsymbol{z}_i, \boldsymbol{z}_j) = S^{(\boldsymbol{x})}(\boldsymbol{x}_i, \boldsymbol{x}_j)\, S^{(Var)}(v_i, v_j)\, S^{(\boldsymbol{\nabla})}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j)$

- Local feature similarity: (Zelnik-manor et al., 2004) (2)

  - Variance:

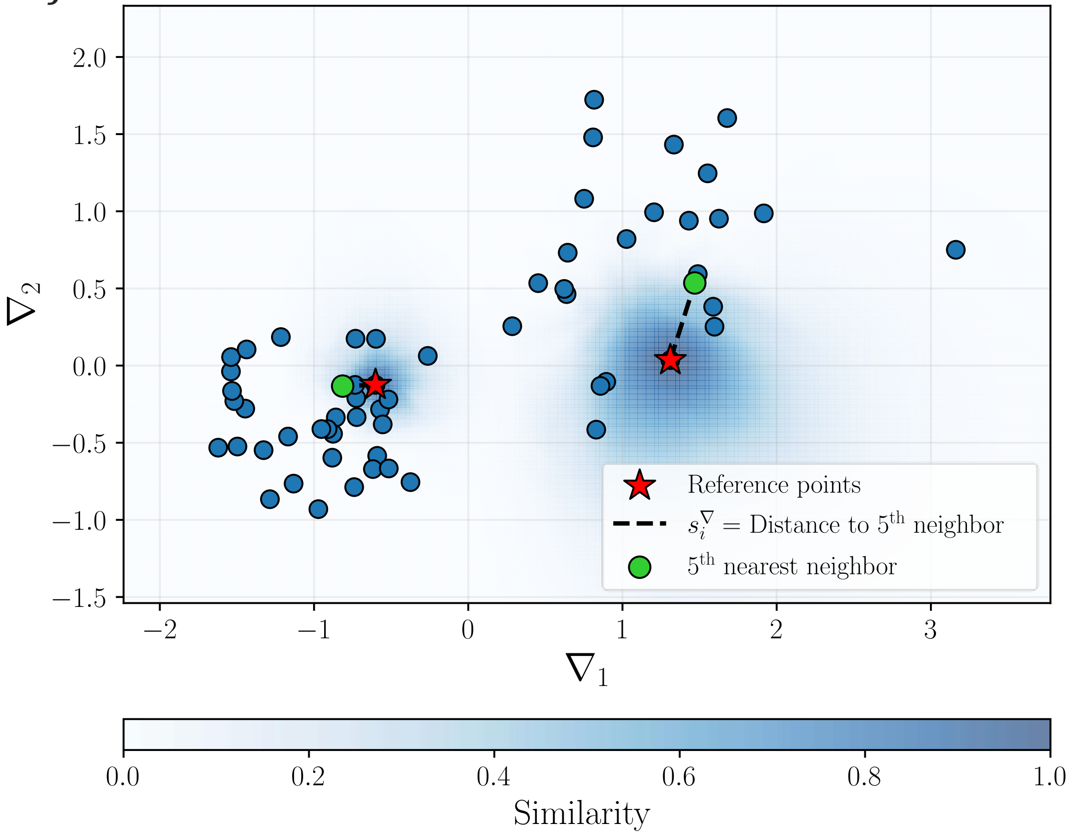
$$S^{(Var)}(v_i, v_j) = \exp\left(-\frac{(v_i - v_j)^2}{s_i^v s_j^v}\right), s_i^{(v)} = \left|v_i - v_i^{(c_v)}\right|$$

$$v_i^{(c_v)} = c_v\text{-th nearest neighbor of } v_i$$

  - Gradient:

$$S^{(\boldsymbol{\nabla})}(\boldsymbol{\nabla}_i, \boldsymbol{\nabla}_j) = \exp\left(-\frac{\left\|\boldsymbol{\nabla}_i - \boldsymbol{\nabla}_j\right\|^2}{s_i^{\boldsymbol{\nabla}} s_j^{\boldsymbol{\nabla}}}\right), s_i^{\boldsymbol{\nabla}} = \left\|\boldsymbol{\nabla}_i - \boldsymbol{\nabla}_i^{(c_\nabla)}\right\|$$

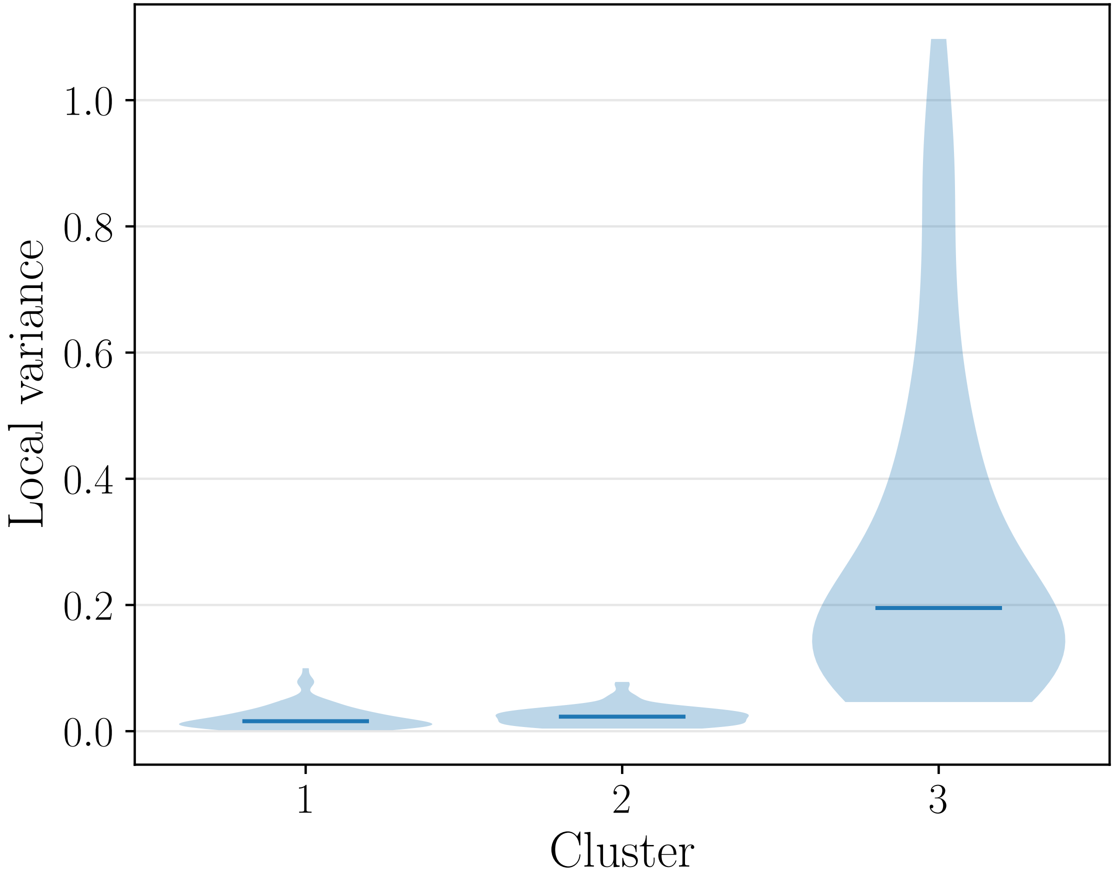
$$\boldsymbol{\nabla}_i^{(c_\nabla)} = c_\nabla\text{-th nearest neighbor of } \boldsymbol{\nabla}_i$$



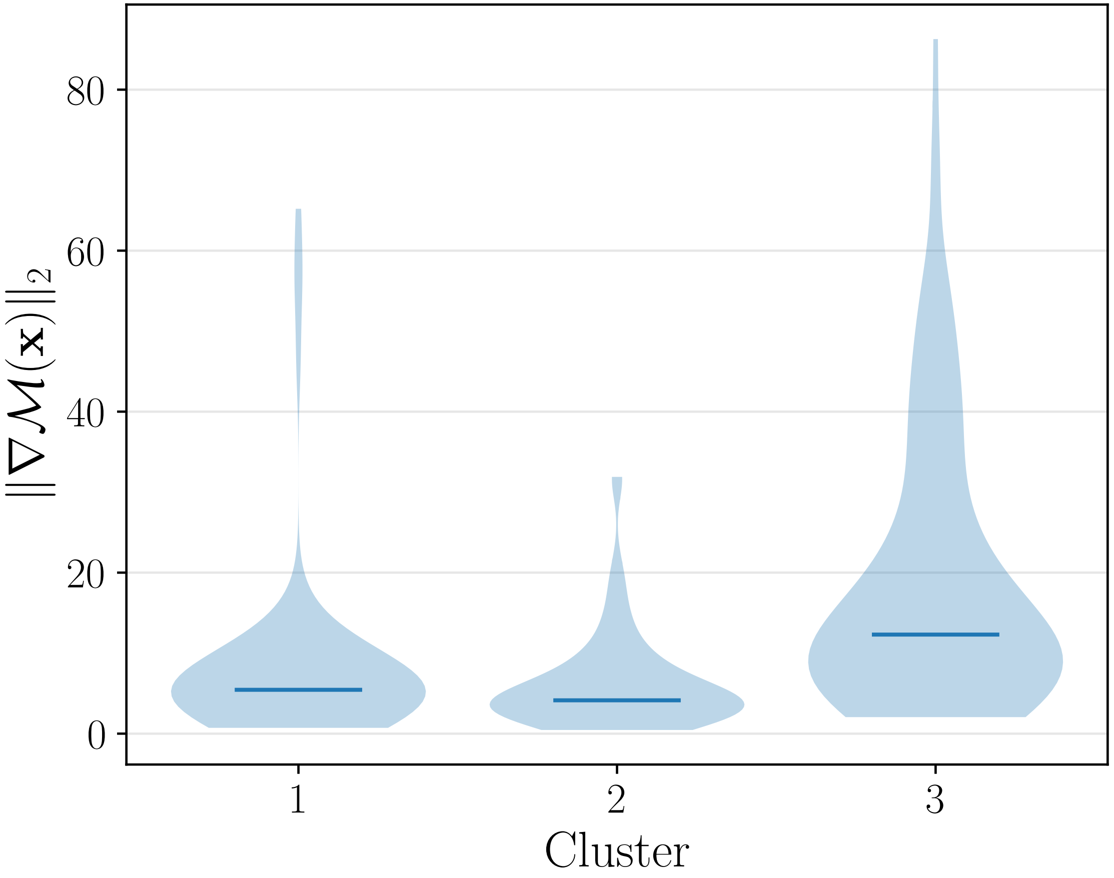
Local scaling for 2D gradients

Clustering results

- **3 clusters** found:



Local variance per found label



Gradient norm per found label