Enhanced Mode Clustering

Yen-Chi Chen

Larry Wasserman   Christopher Genovese

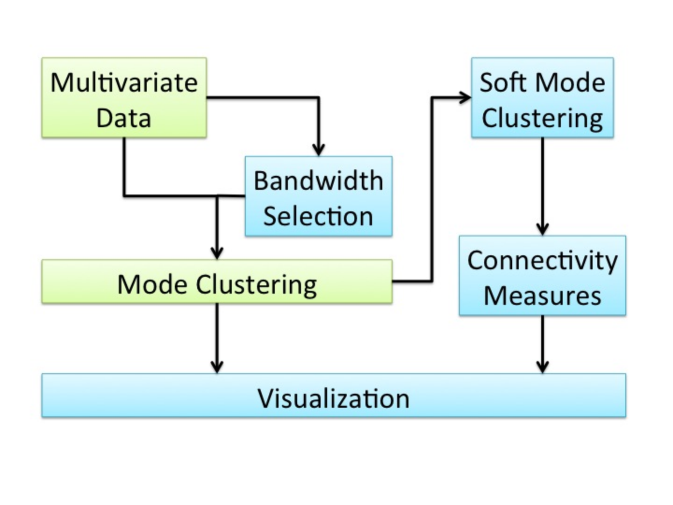
Department of Statistics
Carnegie Mellon University

May 22, 2014

Outline

- Introduction
- Proposed Methods:
  - Soft Mode Clustering
  - Connectivity Measures
  - Bandwidth Selection
  - Visualizations
- Data Analysis
- Conclusion

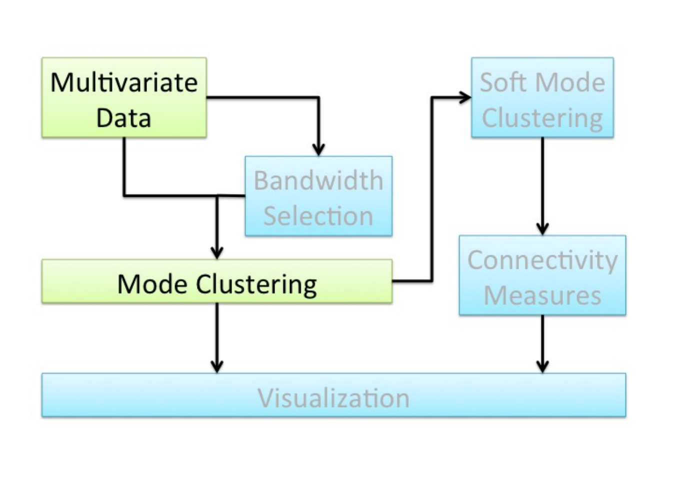
Outline for the Proposed Methods



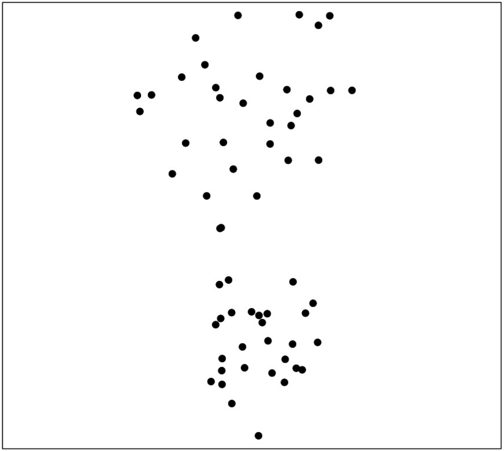
Outline

- Introduction
- Proposed Methods:
  - Soft Mode Clustering
  - Connectivity Measures
  - Bandwidth Selection
  - Visualizations
- Data Analysis
- Conclusion

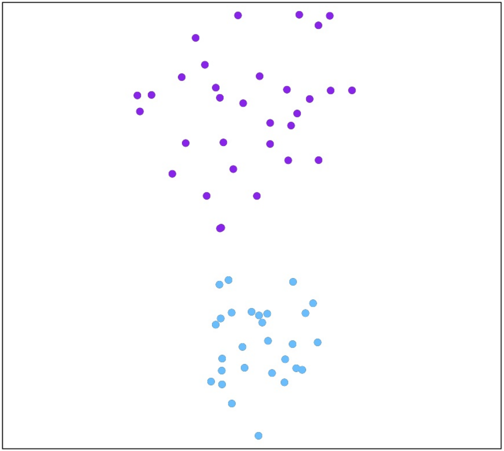
Outline for the Proposed Methods



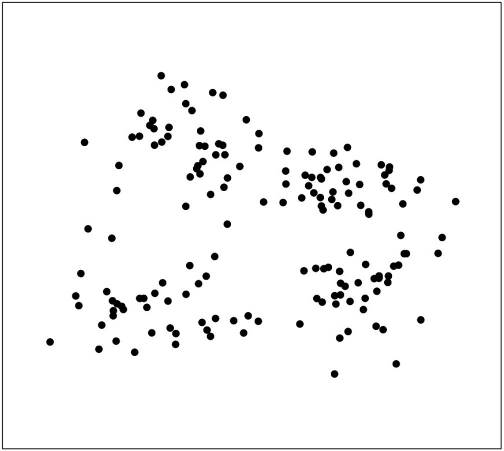
The Goal of Clustering



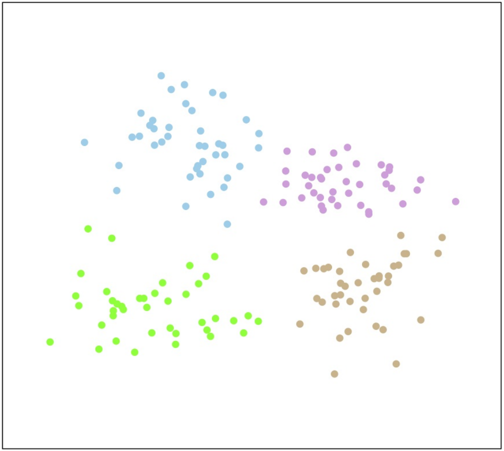
The Goal of Clustering



The Goal of Clustering



The Goal of Clustering



Density Mode Clustering: A Population Level Clustering

- Let $p : \mathbb{R}^d \mapsto \mathbb{R}$ be a density function.
- The gradient $g(x) = \nabla p(x)$ and the Hessian $H(x) = \nabla\nabla p(x)$.

Density Mode Clustering: A Population Level Clustering

- Let $p : \mathbb{R}^d \mapsto \mathbb{R}$ be a density function.
- The gradient $g(x) = \nabla p(x)$ and the Hessian $H(x) = \nabla\nabla p(x)$.
- For each $x \in \mathbb{R}^d$, we construct a flow $\phi : [0, \infty] \mapsto \mathbb{R}^d$ s.t.

$$\phi_x(0) = x, \qquad \phi'_x(t) = g(\phi_x(t)).$$

Density Mode Clustering: A Population Level Clustering

- Let $p : \mathbb{R}^d \mapsto \mathbb{R}$ be a density function.
- The gradient $g(x) = \nabla p(x)$ and the Hessian $H(x) = \nabla\nabla p(x)$.
- For each $x \in \mathbb{R}^d$, we construct a flow $\phi : [0, \infty] \mapsto \mathbb{R}^d$ s.t.

$$\phi_x(0) = x, \qquad \phi_x'(t) = g(\phi_x(t)).$$

- By Morse theory, $\lim_{t\to} \phi_t(t) \in \mathcal{M}$, where

$$\mathcal{M} = \{x : \nabla p(x) = 0, H(x) \text{ negative definite}\}$$

is the set of local modes.

Density Mode Clustering: A Population Level Clustering

- Let $p : \mathbb{R}^d \mapsto \mathbb{R}$ be a density function.
- The gradient $g(x) = \nabla p(x)$ and the Hessian $H(x) = \nabla\nabla p(x)$.
- For each $x \in \mathbb{R}^d$, we construct a flow $\phi : [0, \infty] \mapsto \mathbb{R}^d$ s.t.

$$\phi_x(0) = x, \qquad \phi_x'(t) = g(\phi_x(t)).$$

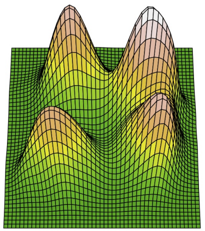
- By Morse theory, $\lim_{t\to} \phi_t(t) \in \mathcal{M}$, where

$$\mathcal{M} = \{x : \nabla p(x) = 0, H(x) \text{ negative definite}\}$$

  is the set of local modes.
- We denote $\mathcal{M} = \{m_1, \cdots, m_k\}$.

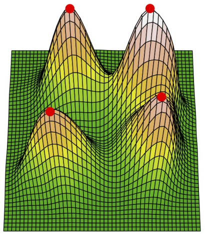
Density Mode Clustering: An Example

- Given a smooth function.



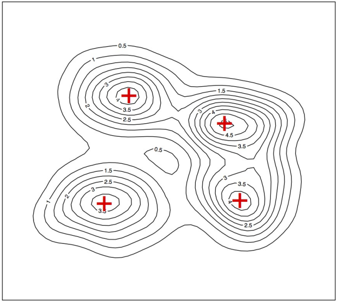
Density Mode Clustering: An Example

- Given a smooth function.
- Find the local modes.



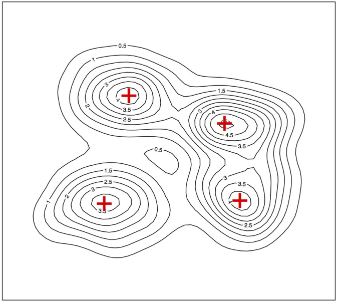
Density Mode Clustering: An Example

- Given a smooth function.
- Find the local modes.



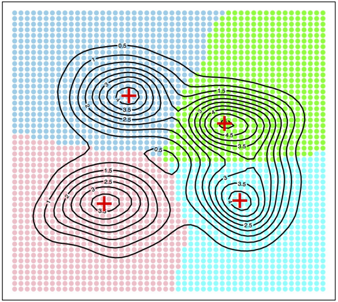
Density Mode Clustering: An Example

- Given a smooth function.
- Find the local modes.
- Following the gradient until arriving a mode.



Density Mode Clustering: An Example

- Given a smooth function.
- Find the local modes.
- Following the gradient until arriving a mode.



Density Mode Clustering: Based on the KDE

- The kernel density estimator (KDE):

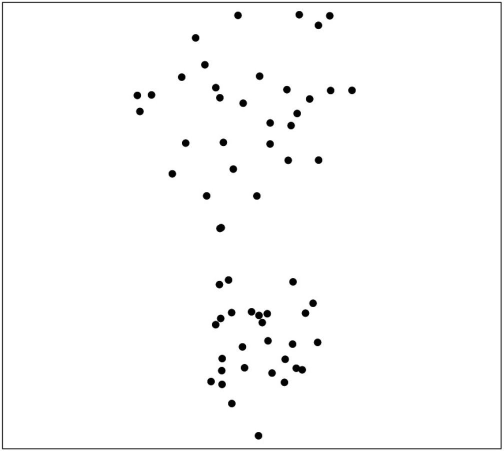
$$\widehat{p}_n(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - X_i}{h}\right).$$

- The gradient:

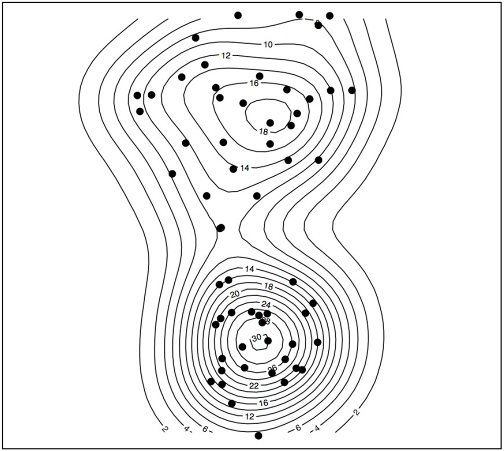
$$\widehat{g}_n(x) = \nabla \widehat{p}_n(x) = \frac{1}{nh^d} \sum_{i=1}^{n} \nabla K\left(\frac{x - X_i}{h}\right).$$

- Clustering: Based on the gradient of $\widehat{g}_n(x)$.
- Algorithm: The mean shift algorithm [Fukunaga1975, Cheng1995, Comaniciu2002].

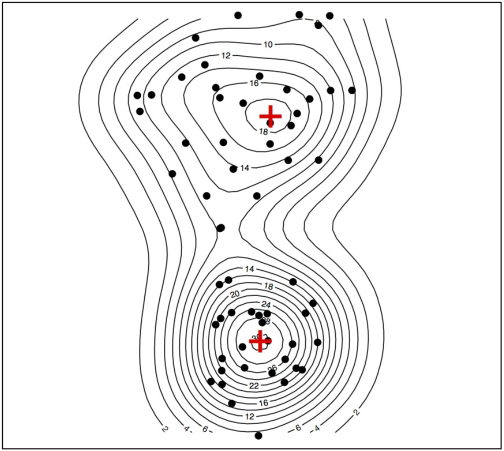
Density Mode Clustering: Based on the KDE



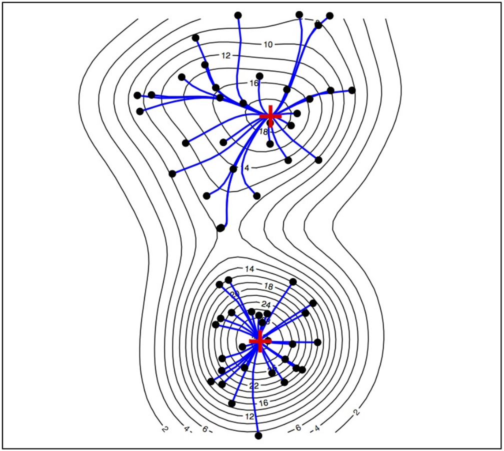
Density Mode Clustering: Based on the KDE



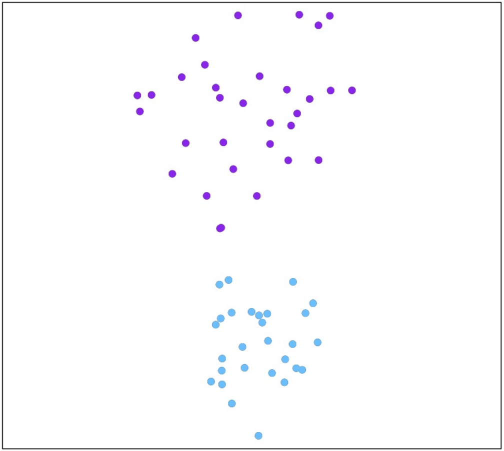
Density Mode Clustering: Based on the KDE



Density Mode Clustering: Based on the KDE



Density Mode Clustering: Based on the KDE



Conventions on Notations

- True local modes

$$\mathcal{M} = \{m_1, \cdots, m_k\}.$$

- Estimated local modes

$$\widehat{\mathcal{M}}_n = \{\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}\}.$$

- The cluster regions (also known as basins of attraction):

$$C_j = \{x : x \text{ being assigned to } m_j \text{ under } g\}.$$

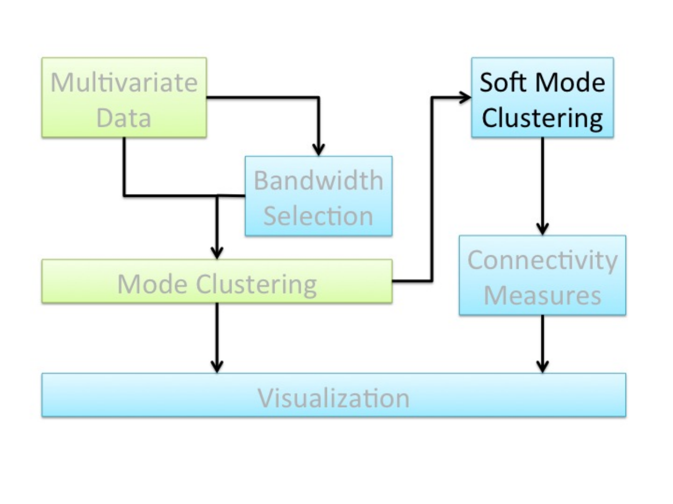
- The estimated cluster regions:

$$\widehat{C}_j = \{x : x \text{ being assigned to } \widehat{m}_j \text{ under } \widehat{g}_n\}.$$

Outline

- Introduction
- Proposed Methods:
  - Soft Mode Clustering
  - Connectivity Measures
  - Bandwidth Selection
  - Visualizations
- Data Analysis
- Conclusion

Outline for the Proposed Methods



Basic Ideas for Soft Clustering

- Usual (Hard) clustering: assign each data to a cluster.
  e.g. $a(x) = (0, 1, 0, 0, 0)$: assign x to the second cluster.

Basic Ideas for Soft Clustering

- Usual (Hard) clustering: assign each data to a cluster.
  e.g. $a(x) = (0, 1, 0, 0, 0)$: assign x to the second cluster.
- Soft clustering: assign each data to a mixture of clusters.
  e.g. $a(x) = (0.05, 0.7, 0.2, 0.05, 0)$:

Basic Ideas for Soft Clustering

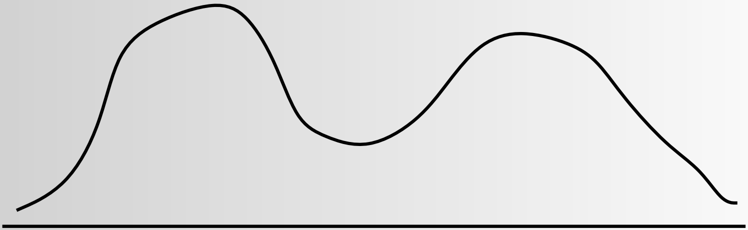
- Usual (Hard) clustering: assign each data to a cluster.
  e.g. $a(x) = (0, 1, 0, 0, 0)$: assign x to the second cluster.
- Soft clustering: assign each data to a mixture of clusters.
  e.g. $a(x) = (0.05, 0.7, 0.2, 0.05, 0)$:
  $\longrightarrow$ We have strong **confidence** that x is assinged to cluster 2

Soft Mixture Clustering

- A common soft clustering method: mixture model.
- $p(x) = \pi p_1(x) + (1-\pi)p_2(x)$
- But this is ill-defined.

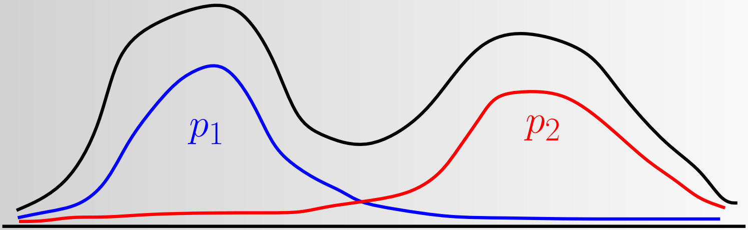
Soft Mixture Clustering

- A common soft clustering method: mixture model.
- $p(x) = \pi p_1(x) + (1 - \pi) p_2(x)$
- But this is ill-defined.



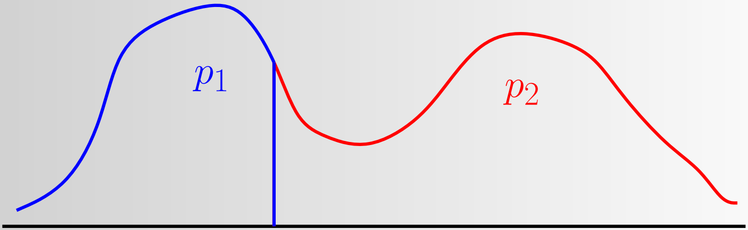
Soft Mixture Clustering

- A common soft clustering method: mixture model.
- $p(x) = \pi p_1(x) + (1-\pi)p_2(x)$
- But this is ill-defined.



Soft Mixture Clustering

- A common soft clustering method: mixture model.
- $p(x) = \pi p_1(x) + (1 - \pi) p_2(x)$
- But this is ill-defined.



Basic Ideas for Soft Mode Clustering

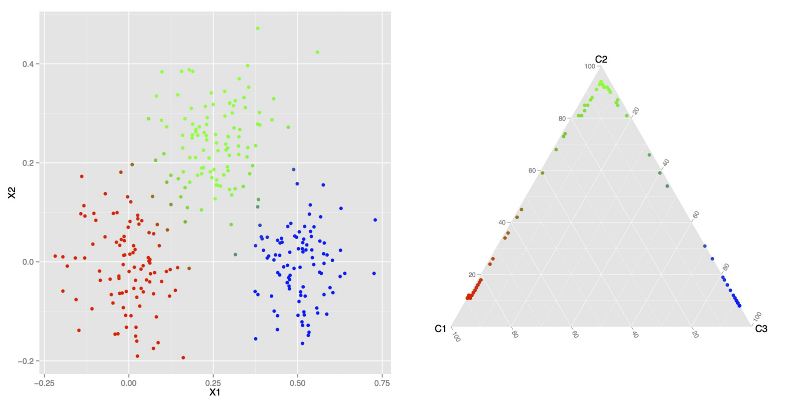
- In mode clustering, we have fixed local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
- All we need is to construct the **soft assignment vector** $a(x)$.

Soft Mode Clustering: The Bootstrap

1. Given data points $X_1, \cdots, X_n$, we find the local modes.
2. For each $x \in \mathbb{R}$, perform the bootstrap and redo the mode clusteirng.
3. Construct the soft assignment vector $a(x) = \left(a_1, \cdots, a_{\widehat{k}}(x)\right)$ where

$$a_\ell(x) = \text{ fraction of } x \text{ being assigned to cluster } \ell.$$

The Bootstrap: Example



Soft Mode Clustering: The Hitting Probability

- We define a diffusion between local modes and data points.
- $\widehat{k} + n$ states: $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}, X_1, \cdots, X_n$.

Soft Mode Clustering: The Hitting Probability

- We define a diffusion between local modes and data points.
- $\widehat{k}+n$ states: $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}, X_1, \cdots, X_n$.
- The first K states: absorbing states.

Soft Mode Clustering: The Hitting Probability

- We define a diffusion between local modes and data points.
- $\widehat{k}+n$ states: $\widehat{m}_1,\cdots,\widehat{m}_{\widehat{k}},X_1,\cdots,X_n$.
- The first K states: absorbing states.
- The transition probability between data points:

$$\mathbf{P}(X_i \to X_j) = \frac{K\left(\frac{X_i-X_j}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{X_i-X_j}{h}\right) + \sum_{\ell=1}^{\widehat{k}} K\left(\frac{X_i-\widehat{m}_\ell}{h}\right)}.$$

- The transition probability to local modes:

$$\mathbf{P}(X_i \to \widehat{m}_\ell) = \frac{K\left(\frac{X_i-m_\ell}{h}\right)}{\sum_{j=1}^{n} K\left(\frac{X_i-X_j}{h}\right) + \sum_{\ell=1}^{\widehat{k}} K\left(\frac{X_i-\widehat{m}_\ell}{h}\right)}.$$

Soft Mode Clustering: The Hitting Probability

- We define a diffusion between local modes and data points.
- $\widehat{k}+n$ states: $\widehat{m}_1,\cdots,\widehat{m}_{\widehat{k}},X_1,\cdots,X_n$.
- The first K states: absorbing states.
- The transition probability between data points:

$$\mathbf{P}(X_i\to X_j)=\frac{K\left(\frac{X_i-X_j}{h}\right)}{\sum_{j=1}^n K\left(\frac{X_i-X_j}{h}\right)+\sum_{\ell=1}^{\widehat{k}} K\left(\frac{X_i-\widehat{m}_\ell}{h}\right)}.$$

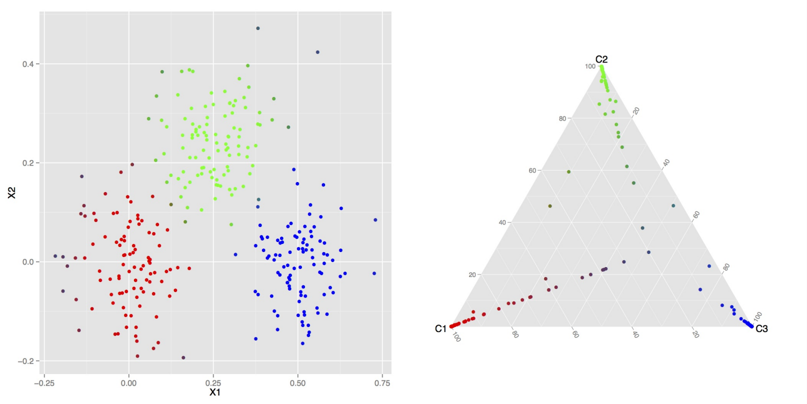
- The transition probability to local modes:

$$\mathbf{P}(X_i\to \widehat{m}_\ell)=\frac{K\left(\frac{X_i-m_\ell}{h}\right)}{\sum_{j=1}^n K\left(\frac{X_i-X_j}{h}\right)+\sum_{\ell=1}^{\widehat{k}} K\left(\frac{X_i-\widehat{m}_\ell}{h}\right)}.$$

- Soft assignement vector:

$$a_\ell(X_i)=\mathbb{P}\left(\text{from } X_i \text{ and hits } \widehat{m}_\ell \text{ first}\right)$$

The Hitting Probability: Example

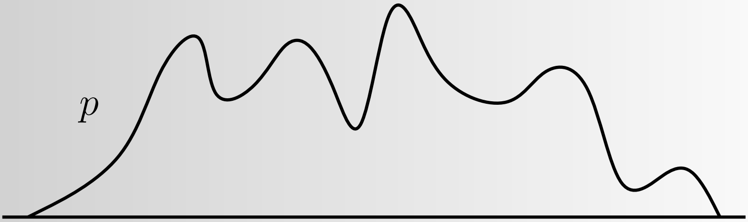


Soft Mode Clustering: The Level Set

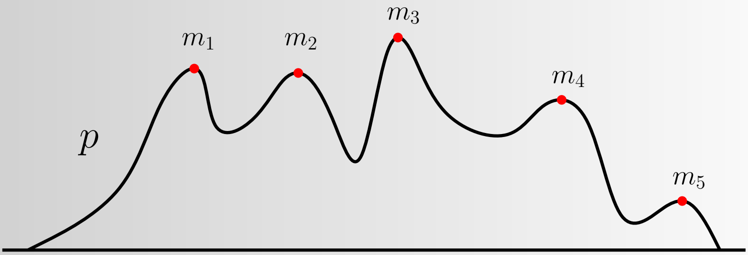
- The third method is based on the level set.
- We create a distance $d_\ell(x)$ for each $\ell = 1, \cdots, k$.
- Transform the distance into soft assignment vector. e.g.

$$a_\ell(x) = \frac{\exp(-\beta_0 d_\ell(x))}{\sum_{j=1}^{k} \exp(-\beta_0 d_j(x))}.$$

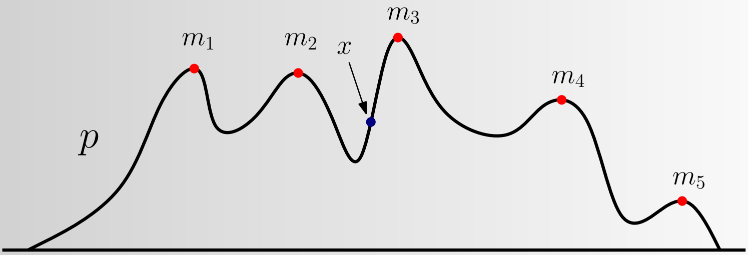
Soft Mode Clustering: The Level Set



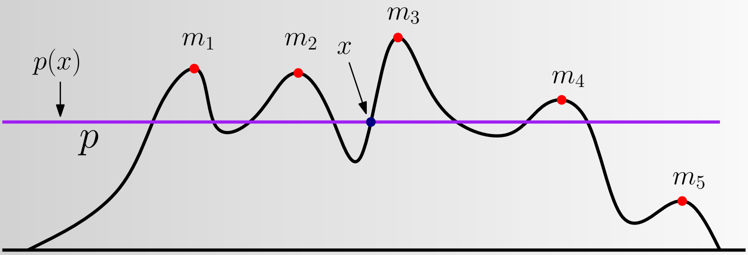
Soft Mode Clustering: The Level Set



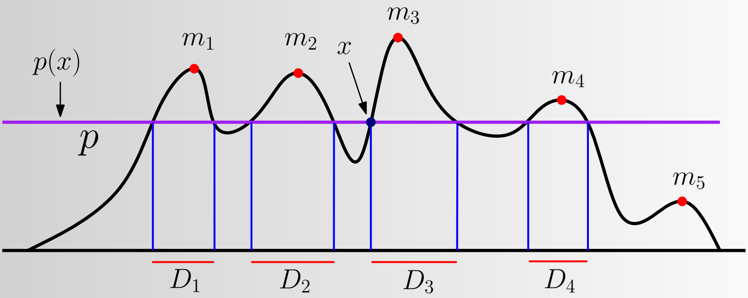
Soft Mode Clustering: The Level Set



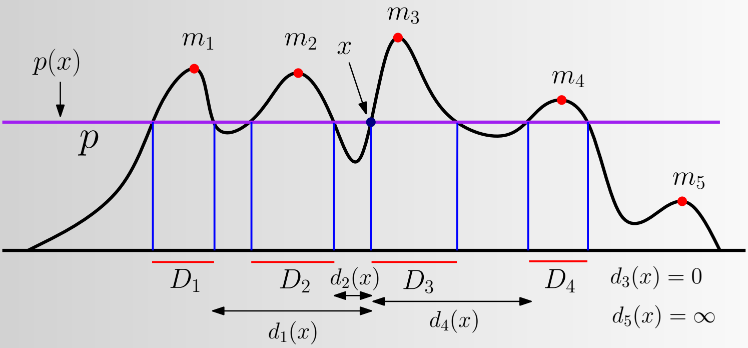
Soft Mode Clustering: The Level Set



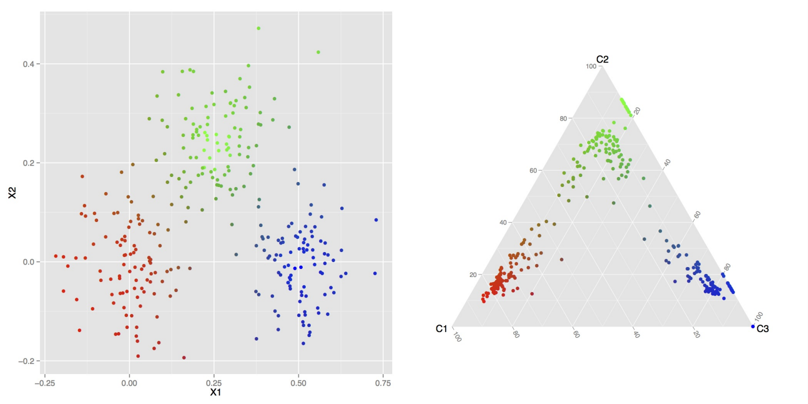
Soft Mode Clustering: The Level Set



Soft Mode Clustering: The Level Set



The Level Set: Example



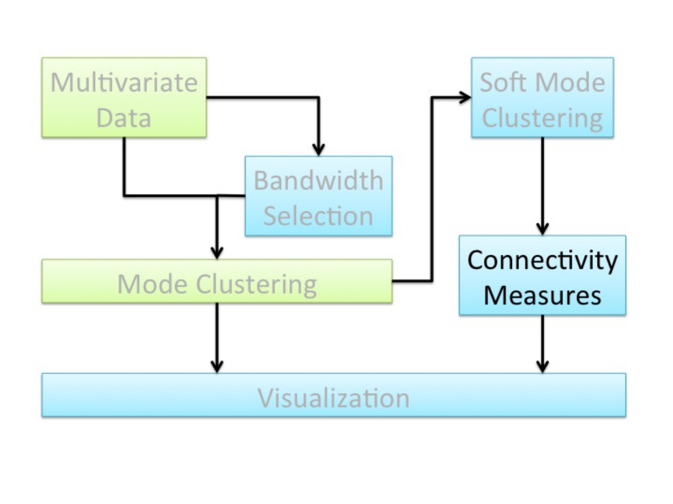
Soft Mode Clustering: Other Distance Methods

Other possible approaches:

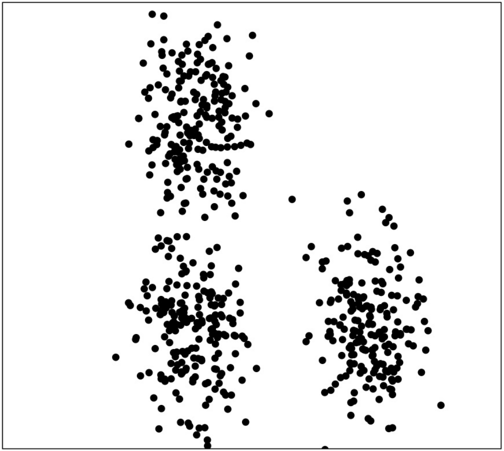
- Diffusion distance
- Density integral distance

We need a conversion between distances $d_\ell(x)$ and soft assignment vector $a(x)$.

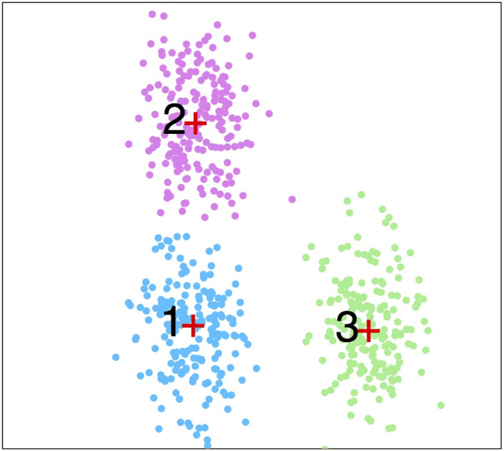
Outline for the Proposed Methods



A Motivating Example



A Motivating Example



Soft Assignment Vector: A Measure of Overlapping

Recall: $\widehat{C}_j$ the regions belong to cluster j.

Soft Assignment Vector: A Measure of Overlapping

Recall: $\widehat{C}_j$ the regions belong to cluster j.

- The soft assignment vector $a(x)$ measures the confidence to be assigned to each cluster.

Soft Assignment Vector: A Measure of Overlapping

Recall: $\widehat{C}_j$ the regions belong to cluster j.

- The soft assignment vector $a(x)$ measures the confidence to be assigned to each cluster.
- $a_\ell(x)$: the confidence of x being assigned to cluster ℓ.

Soft Assignment Vector: A Measure of Overlapping

Recall: $\widehat{C}_j$ the regions belong to cluster j.

- The soft assignment vector $a(x)$ measures the confidence to be assigned to each cluster.
- $a_\ell(x)$: the confidence of x being assigned to cluster ℓ.
- The quantity

$$\frac{1}{N_j}\sum_{i:X_i\in\widehat{C}_j} a_\ell(X_i)$$

  measures the confidence for cluster j being assigned to cluster ℓ; note N_j is the number of points in $\widehat{C}_j$.

Soft Assignment Vector: A Measure of Overlapping

Recall: $\widehat{C}_j$ the regions belong to cluster j.

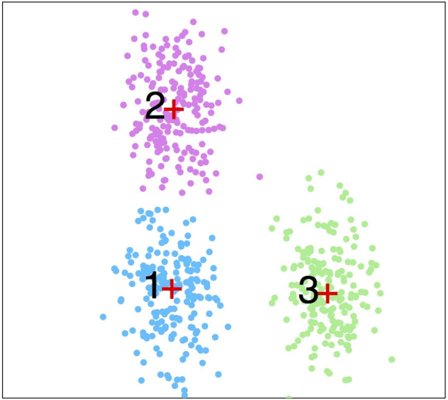
- The soft assignment vector $a(x)$ measures the confidence to be assigned to each cluster.
- $a_\ell(x)$: the confidence of x being assigned to cluster ℓ.
- The quantity

$$\frac{1}{N_j}\sum_{i:X_i\in\widehat{C}_j} a_\ell(X_i)$$

  measures the confidence for cluster j being assigned to cluster ℓ; note N_j is the number of points in $\widehat{C}_j$.
- We define the *connectivity measure* between cluster j,ℓ as

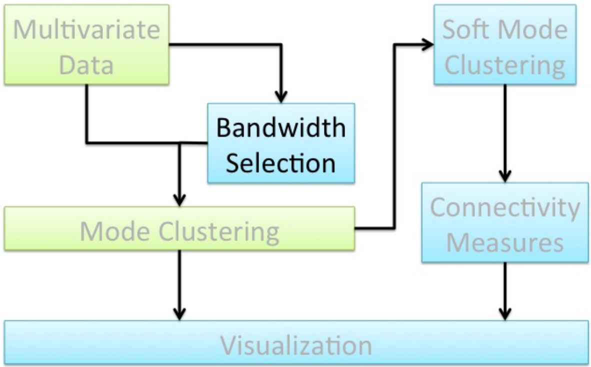
$$\Omega_{j\ell}=\frac{1}{2}\left(\frac{1}{N_j}\sum_{i:X_i\in\widehat{C}_j} a_\ell(X_i)+\frac{1}{N_\ell}\sum_{i:X_i\in\widehat{C}_\ell} a_j(X_i)\right).$$

Example for Connectivity Matrix



	1	2	3
1	–	0.27	0.21
2	0.27	–	0.12
3	0.21	0.12	–

Outline for the Proposed Methods



Optimality for Bandwidth

- Usually, we select smoothing bandwidth h according to minimize some loss function.
- Mean integrated square errors (MISE):

$$MISE(\widehat{p}_n) = \mathbb{E}\left(\int (\widehat{p}_n(x) - p(x))^2\, dx\right).$$

- L_∞ loss:

$$\|\widehat{p}_n - p\|_\infty = \sup_x |\widehat{p}_n(x) - p(x)|.$$

Optimality for Mode Clustering

For mode clustering, the important quantity is the gradient $g(x)$ and its estimator $\widehat{g}_n(x)$.

- MISE:

$$MISE(\widehat{g}_n) = \mathbb{E}\left(\int \|\widehat{g}_n(x) - g(x)\|_2^2\, dx\right).$$

- L_∞ loss:

$$\|\widehat{g}_n - g\|_\infty = \sup_x \|\widehat{g}_n(x) - g(x)\|_{\max}.$$

Rate of Convergence and Bandwidth Selection

- MISE:

$$MISE(\widehat{g}_n) = O(h^4) + O\left(\frac{1}{nh^{d+2}}\right).$$

Rate of Convergence and Bandwidth Selection

- MISE:

$$MISE(\widehat{g}_n) = O(h^4) + O\left(\frac{1}{nh^{d+2}}\right).$$

- L_∞ loss:

$$\|\widehat{g}_n - g\|_\infty = O(h^2) + O_P\left(\sqrt{\frac{\log n}{nh^{d+2}}}\right).$$

Rate of Convergence and Bandwidth Selection

- MISE:

$$MISE(\widehat{g}_n) = O(h^4) + O\left(\frac{1}{nh^{d+2}}\right).$$

- L_∞ loss:

$$\|\widehat{g}_n - g\|_\infty = O(h^2) + O_P\left(\sqrt{\frac{\log n}{nh^{d+2}}}\right).$$

- This suggests two different optimality criteria:

$$h_{MISE} = C_1\left(\frac{1}{n}\right)^{\frac{1}{d+6}} \qquad h_{L_\infty} = C_2\left(\frac{\log n}{n}\right)^{\frac{1}{d+6}}$$

Rate of Convergence and Bandwidth Selection

- MISE:

$$MISE(\widehat{g}_n) = O(h^4) + O\left(\frac{1}{nh^{d+2}}\right).$$

- L_∞ loss:

$$\|\widehat{g}_n - g\|_\infty = O(h^2) + O_P\left(\sqrt{\frac{\log n}{nh^{d+2}}}\right).$$

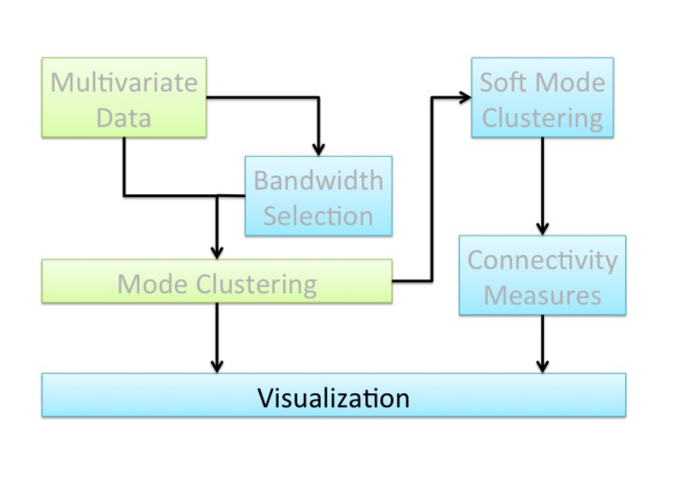
- This suggests two different optimality criteria:

$$h_{MISE} = C_1 \left(\frac{1}{n}\right)^{\frac{1}{d+6}} \qquad h_{L_\infty} = C_2 \left(\frac{\log n}{n}\right)^{\frac{1}{d+6}}$$

- In practice, we use the normal reference rule [Sliverman1986, Chacon2011,13]:

$$h_{NR} = sd(\mathbb{X}) \times \left(\frac{4}{d+4}\right)^{\frac{1}{d+6}} \left(\frac{1}{n}\right)^{\frac{1}{d+6}}.$$

Outline for the Proposed Methods



Multidimensional Scaling (MDS): An Introduction

- Input: $X_1, \cdots, X_n \in \mathbb{R}^d$.
- Output: $Z_1, \cdots, Z_n \in \mathbb{R}^r$ with $r < d$.
- Distance preserved:

$$\min \sum_{i \neq j} |d(X_i, X_j) - d(Z_i, Z_j)|$$

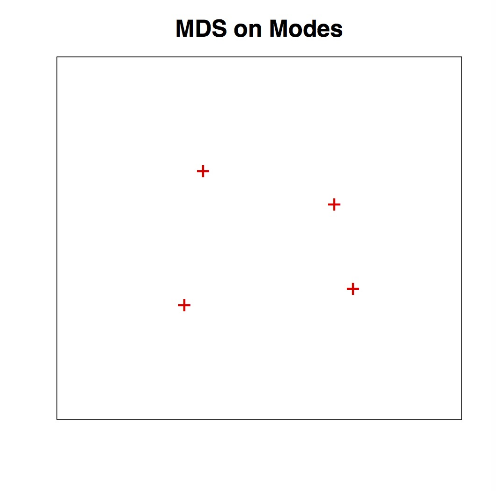
  for some distance function d.
- In practice, we use the classical scaling.

Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.

Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.

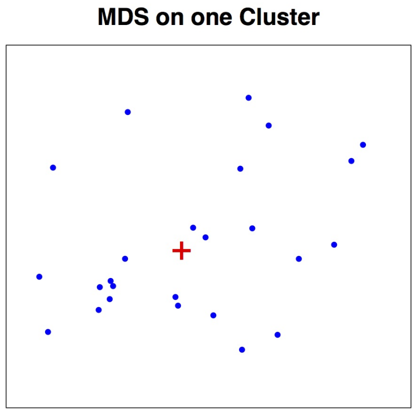


Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.

Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.

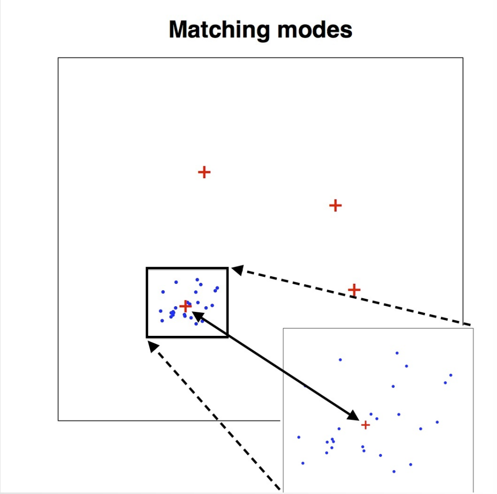


Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.
3. By matching the local modes, we plot cluster points around the mode.

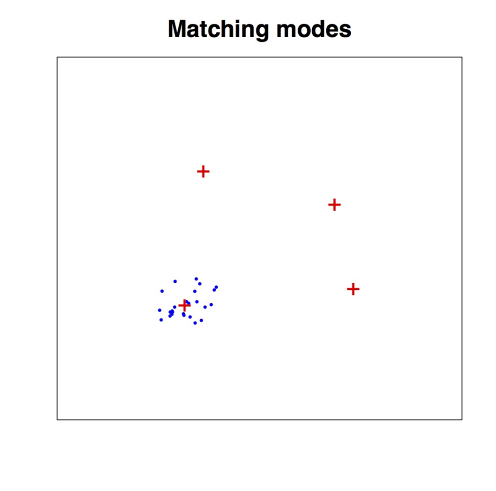
Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.
3. By matching the local modes, we plot cluster points around the mode.



Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.
3. By matching the local modes, we plot cluster points around the mode.

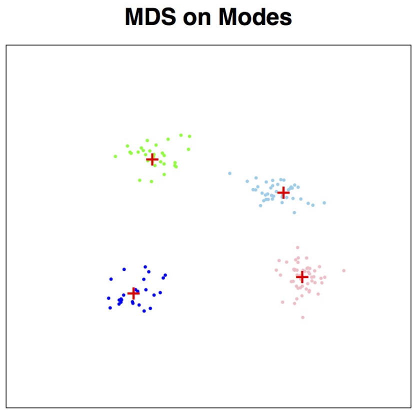


Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.
3. By matching the local modes, we plot cluster points around the mode.
4. Repeat (2-3) to each mode.

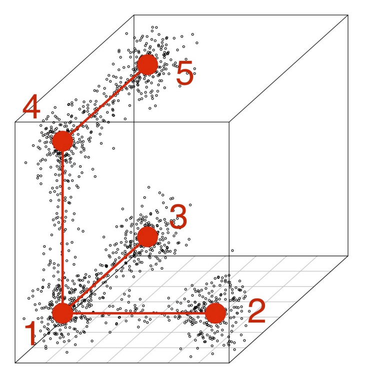
Two-Stage MDS

1. We apply MDS to local modes $\widehat{m}_1, \cdots, \widehat{m}_{\widehat{k}}$.
2. For each cluster, we apply MDS for the cluster points.
3. By matching the local modes, we plot cluster points around the mode.
4. Repeat (2-3) to each mode.



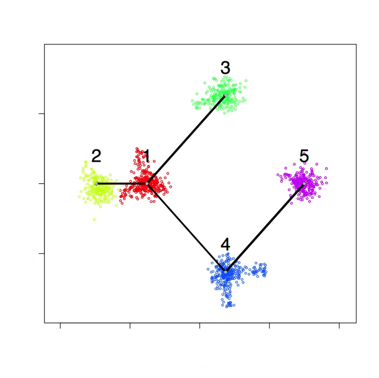
Outline

5-Cluster in d=6



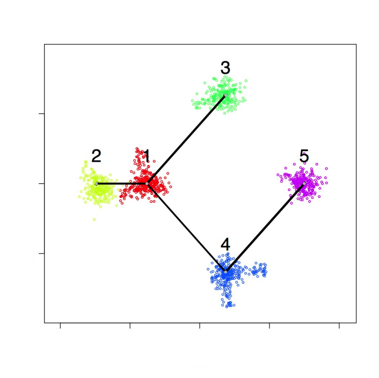
- 5 clusters each with $n_C = 200$.
- 4 edges connecting clusters and each with $n_E = 100$.
- Embedding this structure in $d = 6$ and add Gaussian noise.

5-Cluster in d=6



- 5 clusters each with $n_C = 200$.
- 4 edges connecting clusters and each with $n_E = 100$.
- Embedding this structure in $d = 6$ and add Gaussian noise.

5-Cluster in d=6



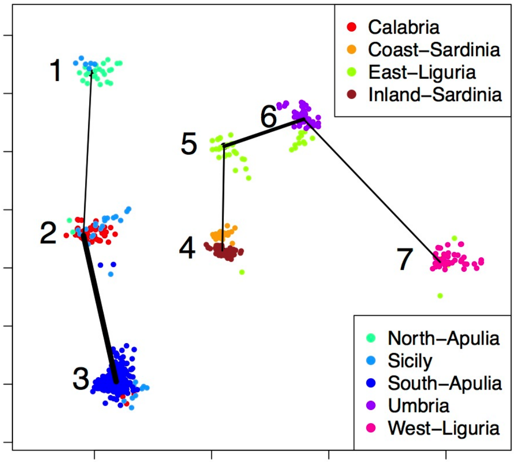
	1	2	3	4	5
1	–	0.17	0.19	0.15	0.05
2	0.17	–	0.05	0.04	0.01
3	0.19	0.05	–	0.05	0.01
4	0.15	0.04	0.05	–	0.20
5	0.05	0.01	0.01	0.20	–

The Olive Oil Data: Description

- A data consists of 572 olive oil sample produced in 9 different areas in Italy.
- We measure 8 different chemical contents for each oil.

	palmitic	palmitoleic	stearic	oleic	linoleic	linolenic	arachidic	eicosenoic
1	1088	73	224	7709	781	31	61	29
2	911	54	246	8113	549	31	63	29
3	966	57	240	7952	619	50	78	35
4	1051	67	259	7771	672	50	80	46
5	911	49	268	7924	678	51	70	44
6	1100	61	235	7728	734	39	64	35

The Olive Oil Data: Analysis



The Olive Oil Data: Analysis

	1	2	3	4	5	6	7
Calabria	0	51	5	0	0	0	0
Coast-Sardinia	0	0	0	33	0	0	0
East-Liguria	0	0	0	1	32	11	6
Inland-Sardinia	0	0	0	65	0	0	0
North-Apulia	23	2	0	0	0	0	0
Sicily	6	19	11	0	0	0	0
South-Apulia	0	2	204	0	0	0	0
Umbria	0	0	0	0	0	51	0
West-Liguria	0	0	0	0	0	0	50

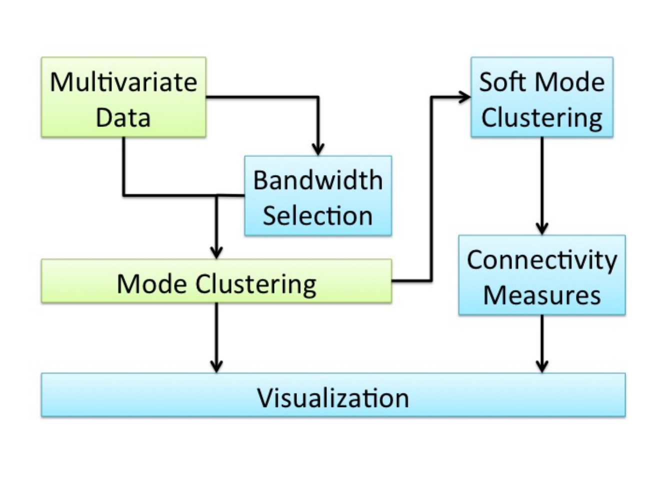
The Olive Oil Data: Analysis

	1	2	3	4	5	6	7
1	–	0.08	0.05	0.00	0.01	0.02	0.00
2	0.08	–	0.30	0.01	0.01	0.00	0.00
3	0.05	0.30	–	0.02	0.01	0.00	0.00
4	0.00	0.01	0.02	–	0.09	0.02	0.01
5	0.01	0.01	0.01	0.09	–	0.19	0.04
6	0.02	0.00	0.00	0.02	0.19	–	0.09
7	0.00	0.00	0.00	0.01	0.04	0.09	–

Outline

- Introduction
- Proposed Methods:
  - Soft Mode Clustering
  - Connectivity Measures
  - Bandwidth Selection
  - Visualizations
- Data Analysis
- Conclusion

Conclusion


Multivariate Data
Bandwidth Selection
Mode Clustering
Soft Mode Clustering
Connectivity Measures
Visualization

Thank you!

reference

1. Y.-C. Chen, C. R. Genovese, L. Wasserman. Asymptotic theory for density ridges, working paper, 2014.
2. Y.-C. Chen, C. R. Genovese, Larry Wasserman. Enhanced Mode Clustering, working paper 2014.
3. J. Chacon and T. Duong. Data-driven density derivative estimation, with applications to nonparametric clustering and bump hunting. Electronic Journal of Statistics, 2013.
4. J. Chacon, T. Duong, and M. Wand. Asymptotics for general multivariate kernel density derivative estimators. Statistica Sinica, 2011.
5. Y. Cheng. Mean shift, mode seeking, and clustering. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 17(8):790799, 1995.
6. D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. Pattern Analysis and Machine Intelligence, IEEE Transactions, 2002.
7. K. Fukunaga and L. D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition, 1975.
8. J. Li, S. Ray, and B. Lindsay. A nonparametric statistical approach to clustering via mode identification. Journal of

   Machine Learning Research, 2007

The Classical Scaling

- Minimizing

$$\sum_{i \neq j} \left| \left(X_i - \bar{X}_n\right)^T \left(X_j - \bar{X}_n\right) - \left(Z_i - \bar{Z}_n\right)^T \left(Z_j - \bar{Z}_n\right) \right|$$

The Classical Scaling

- Minimizing

$$\sum_{i \neq j} \left| (X_i - \bar{X}_n)^T (X_j - \bar{X}_n) - (Z_i - \bar{Z}_n)^T (Z_j - \bar{Z}_n) \right|$$

- Analystical solution:

$$\mathbf{Z} = (Z_1, \cdots, Z_n)^T = \mathbf{V}_k \mathbf{D}_k,$$

The Classical Scaling

- Minimizing

$$\sum_{i \neq j} \left| \left(X_i - \bar{X}_n\right)^T \left(X_j - \bar{X}_n\right) - \left(Z_i - \bar{Z}_n\right)^T \left(Z_j - \bar{Z}_n\right) \right|$$

- Analystical solution:

$$\mathbf{Z} = \left(Z_1, \cdots, Z_n\right)^T = \mathbf{V}_k \mathbf{D}_k,$$

where $\mathbf{V}_k = [v_1, \cdots, v_k]$ and $\mathbf{D}_k = \mathrm{Diag}(\sqrt{(}\lambda_1), \cdots, \sqrt{\lambda_k})$ with (v_j, λ_j) being j-th eigenvector/value of a $n \times n$ matrix $\mathbf{S}$.

The Classical Scaling

- Minimizing

$$\sum_{i \neq j} \left| \left(X_i - \bar{X}_n\right)^T \left(X_j - \bar{X}_n\right) - \left(Z_i - \bar{Z}_n\right)^T \left(Z_j - \bar{Z}_n\right) \right|$$

- Analystical solution:

$$\mathbf{Z} = (Z_1, \cdots, Z_n)^T = \mathbf{V}_k \mathbf{D}_k,$$

where $\mathbf{V}_k = [v_1, \cdots, v_k]$ and $\mathbf{D}_k = \text{Diag}(\sqrt{(}\lambda_1), \cdots, \sqrt{\lambda_k})$ with (v_j, λ_j) being j-th eigenvector/value of a $n \times n$ matrix $\mathbf{S}$.

$\mathbf{S}_{ij} = \left(X_i - \bar{X}_n\right)^T \left(X_j - \bar{X}_n\right)$.