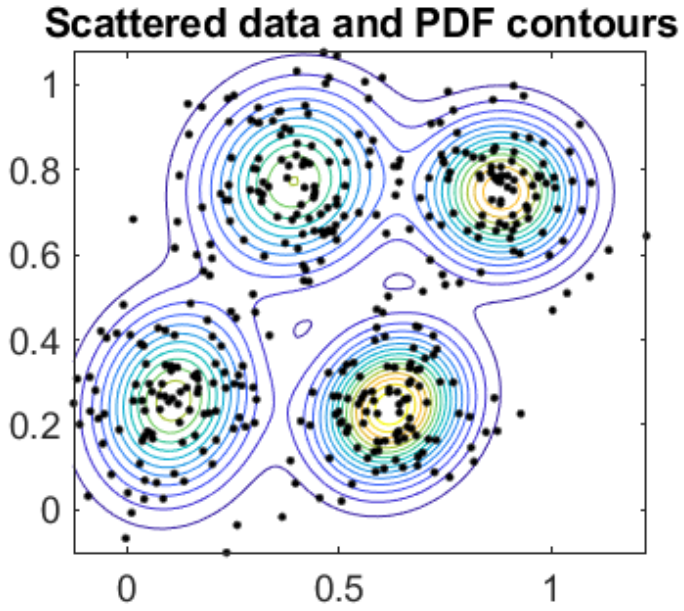
Reinforced EM Algorithm for Clustering with Gaussian Mixture Models through Clever Initialization

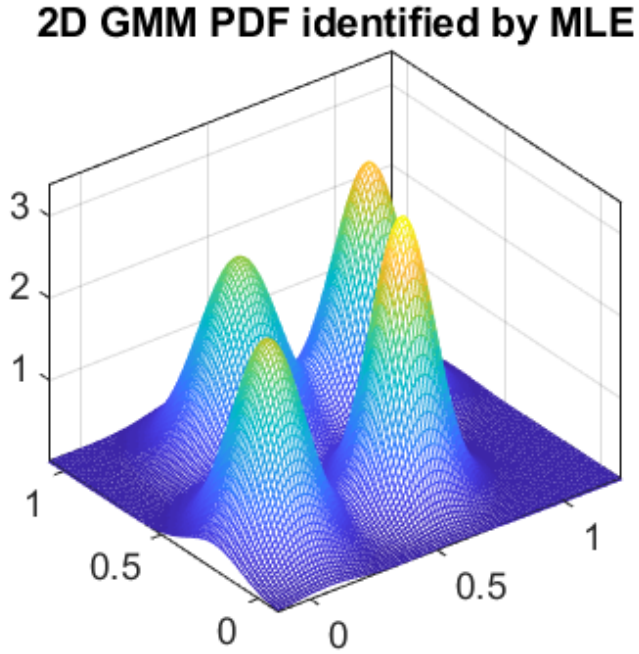
Joshua Tobin
TCD

Chin Pang Ho
CityU HK

Mimi Zhang
TCD

GMM for Clustering





A GMM density has the form

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \phi(\boldsymbol{x}; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j) \,.$$

Clustering is done by assigning each $\boldsymbol{x}_i$ to the mixture component (i.e., cluster) to which it is most likely to belong a posteriori.

EM Algorithm

1. Initialize the parameters: $\{\pi_1, \ldots, \pi_m\}$, $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m\}$ and $\{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m\}$.
2. Compute the responsibilities: for $i = 1, \ldots, n$ and $j = 1, \ldots, m$,

$$r_{ij} = \frac{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{v=1}^{m} \pi_v \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)}.$$

3. Update the estimates: for $j = 1, \ldots, m$,

$$\pi_j = \frac{\sum_{i=1}^{n} r_{ij}}{n}, \quad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} r_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{n} r_{ij}}, \quad \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{n} r_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{n} r_{ij}}.$$

4. Iterate steps 2 and 3 until convergence.

EM Algorithm

With random initialization, converge to bad local maxima with probability $1 - e^{-O(m)}$.

1. Initialize the parameters: $\{\pi_1, \ldots, \pi_m\}$, $\{\boldsymbol{\mu}_1, \ldots, \boldsymbol{\mu}_m\}$ and $\{\boldsymbol{\Sigma}_1, \ldots, \boldsymbol{\Sigma}_m\}$.
2. Compute the responsibilities: for $i = 1, \ldots, n$ and $j = 1, \ldots, m$,

$$r_{ij} = \frac{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}{\sum_{v=1}^{m} \pi_v \phi(\boldsymbol{x}_i; \boldsymbol{\mu}_v, \boldsymbol{\Sigma}_v)}.$$

3. Update the estimates: for $j = 1, \ldots, m$,

$$\pi_j = \frac{\sum_{i=1}^{n} r_{ij}}{n}, \quad \boldsymbol{\mu}_j = \frac{\sum_{i=1}^{n} r_{ij} \boldsymbol{x}_i}{\sum_{i=1}^{n} r_{ij}}, \quad \boldsymbol{\Sigma}_j = \frac{\sum_{i=1}^{n} r_{ij} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^T}{\sum_{i=1}^{n} r_{ij}}.$$

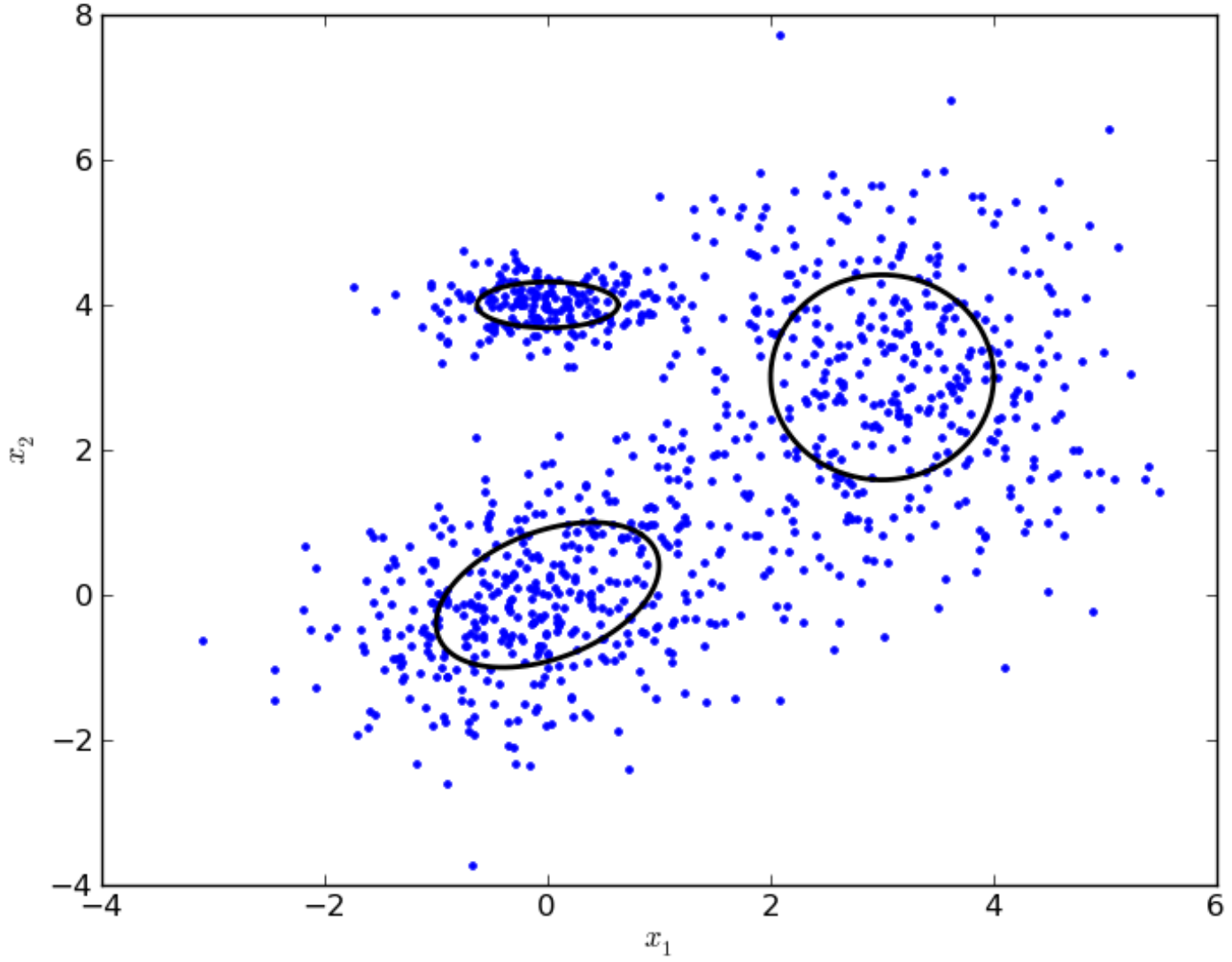
4. Iterate steps 2 and 3 until convergence.

May converge to a singularity.

Exemplar Means

Assume that the clusters are dense enough, such that there is always a data point very close to the real cluster centre.

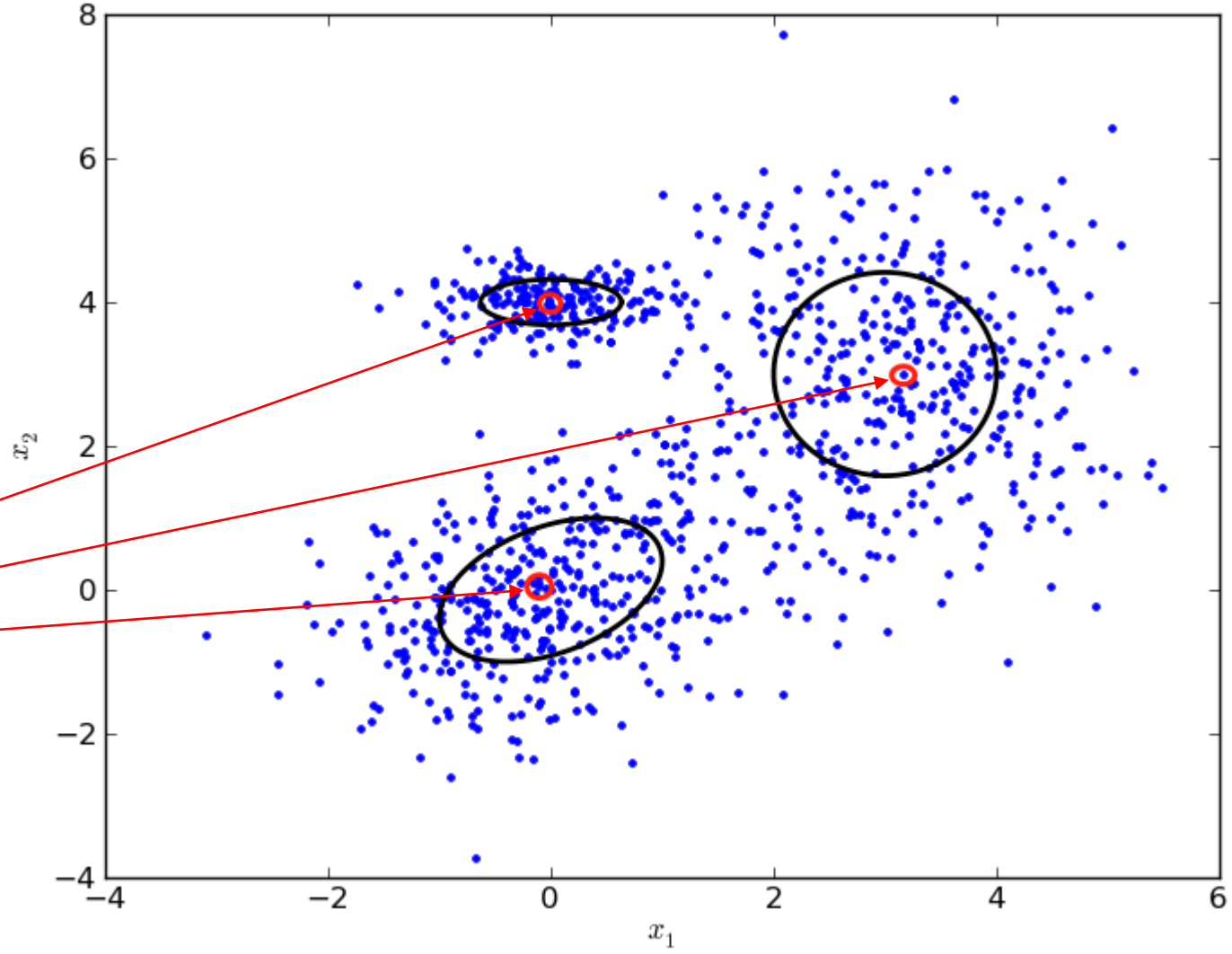
Fix Gaussian means at "exemplars" in the dataset.



Exemplar Means

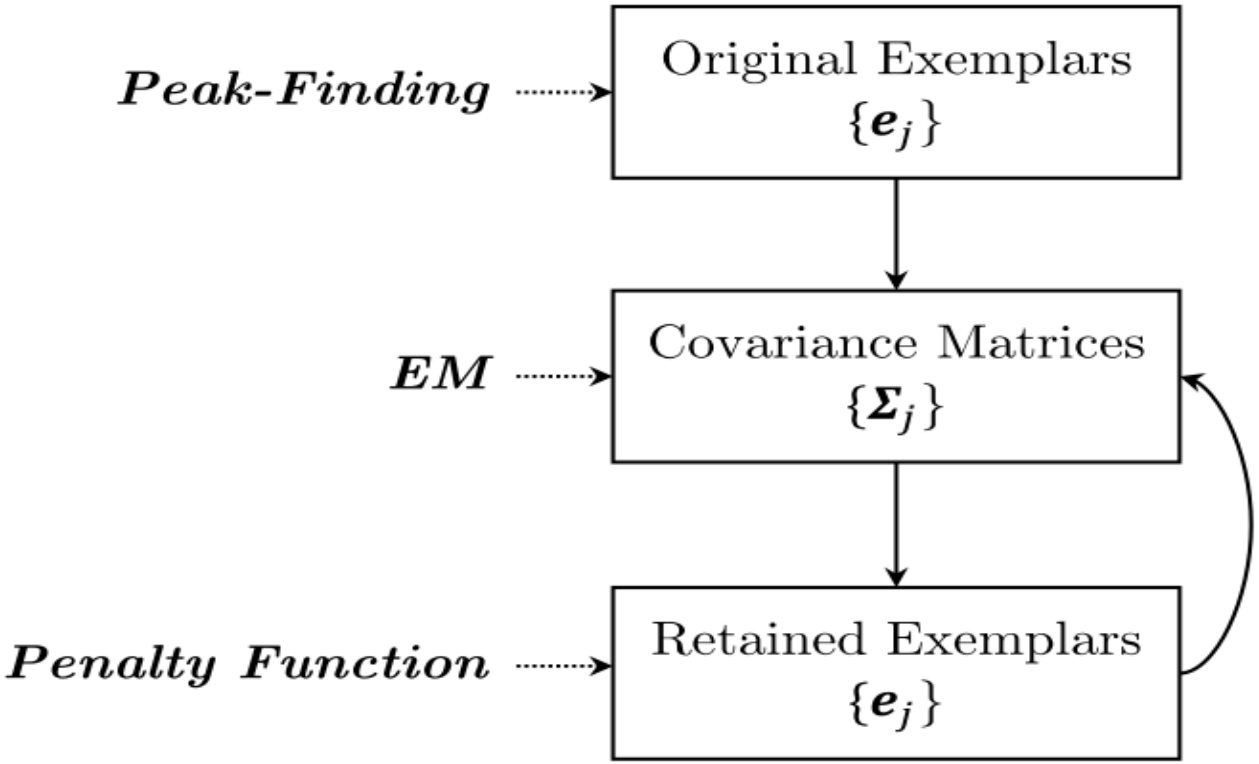
Assume that the clusters are dense enough, such that there is always a data point very close to the real cluster centre.

Fix Gaussian means at "exemplars" in the dataset.



Our Framework

Part 1: Determine an initial set of exemplars $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, guaranteed to include all density peaks (i.e., modes) in the data.



Part 2: Prune redundant exemplars.

Final GMM density has the form

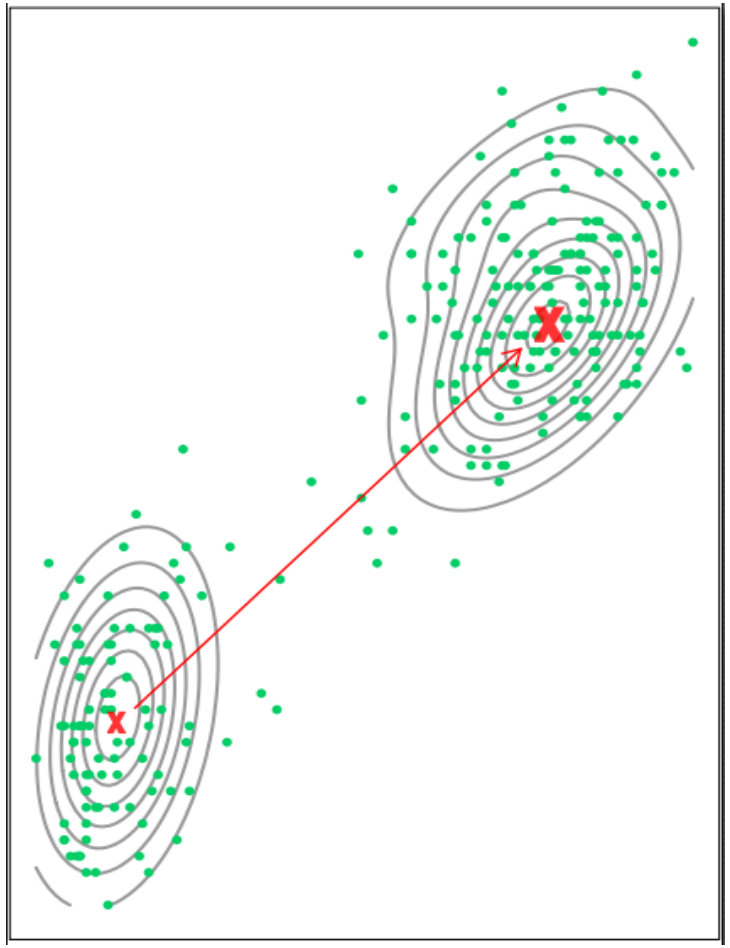
$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \pi_j \phi(\boldsymbol{x}; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j) .$$

Part 1 – Peak Finding

Cluster centres are characterized by:

(1) a higher density than their neighbours

(2) a relatively large distance from points with higher densities

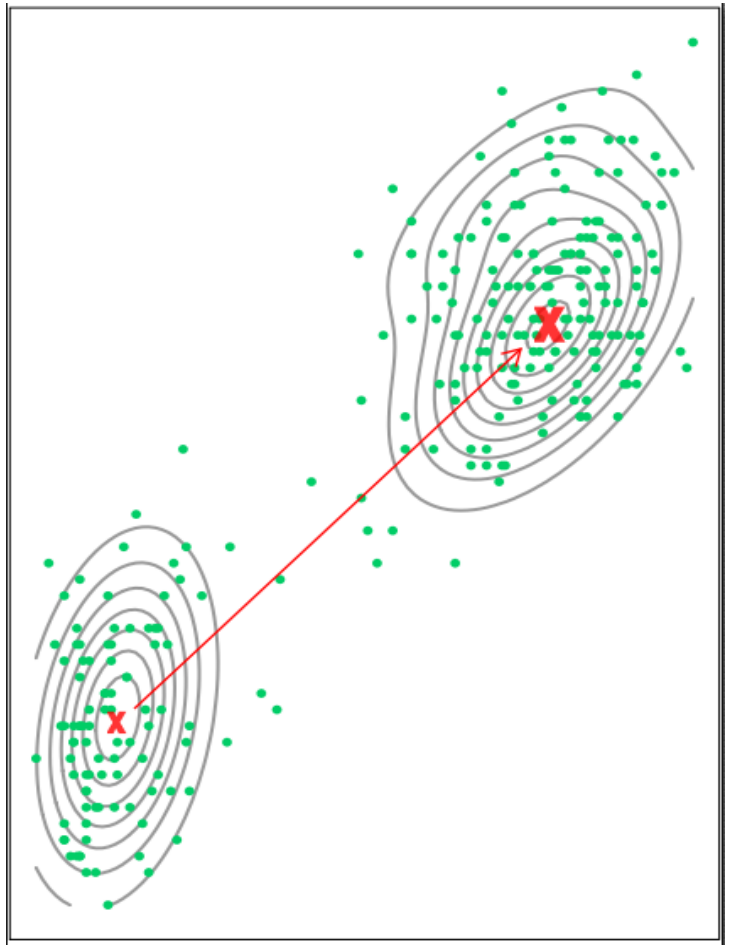


Part 1 – Peak Finding

Cluster centres are characterized by:

(1) a higher density than their neighbours

(2) a relatively large distance from points with higher densities



Part 1 – Peak Finding

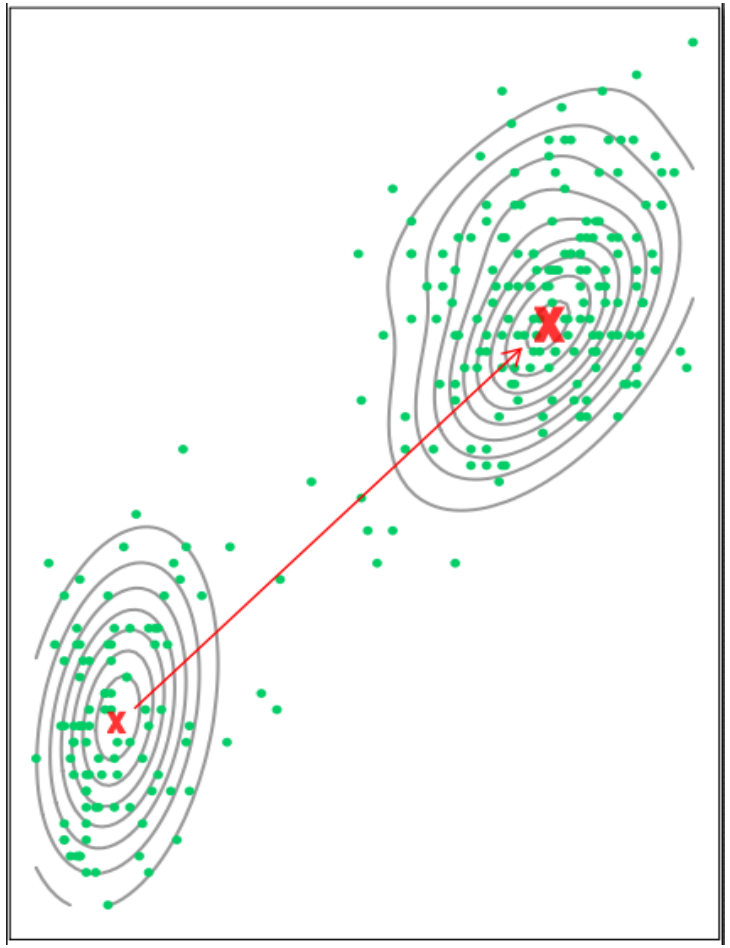
Cluster centres are characterized by:

(1) a higher density than their neighbours

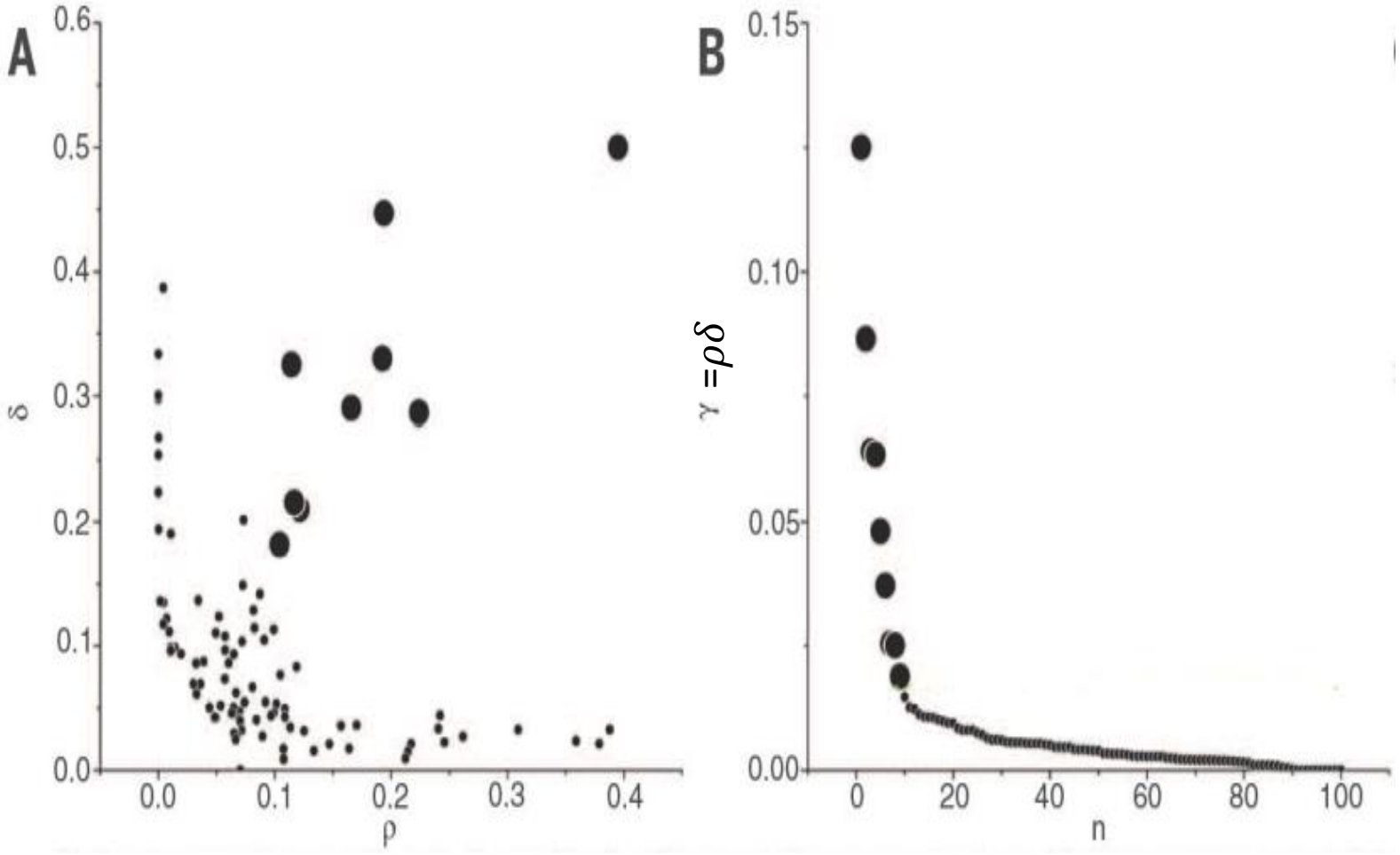
$\rho(\boldsymbol{x})$ – Gaussian kernel density estimate

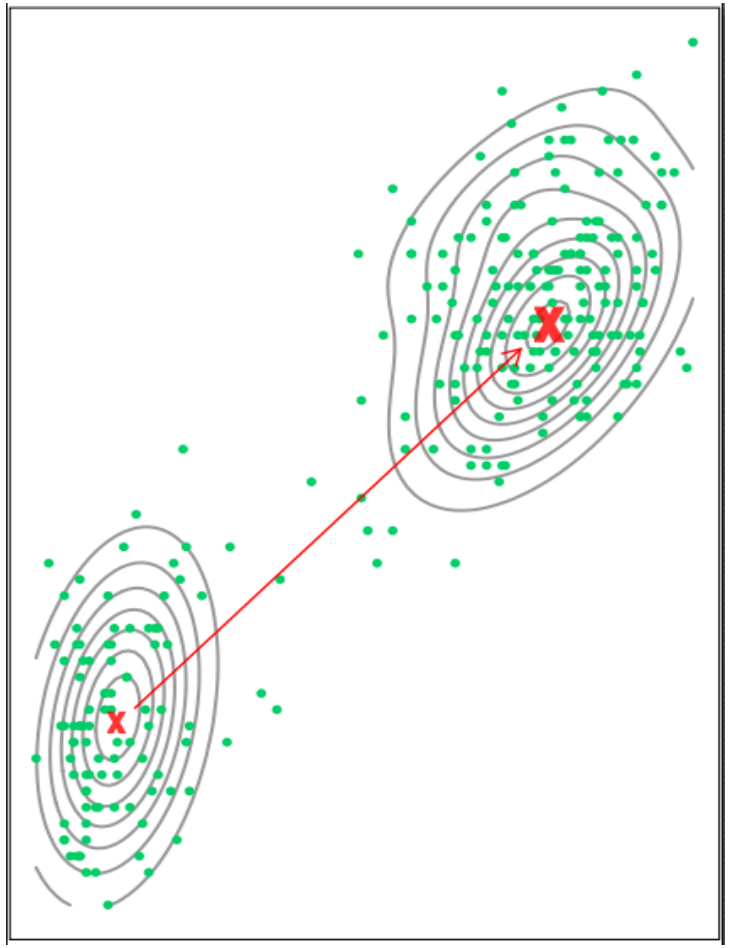
(2) a relatively large distance from points with higher densities

$\delta(\boldsymbol{x})$ -- distance to the nearest neighbour of higher local density

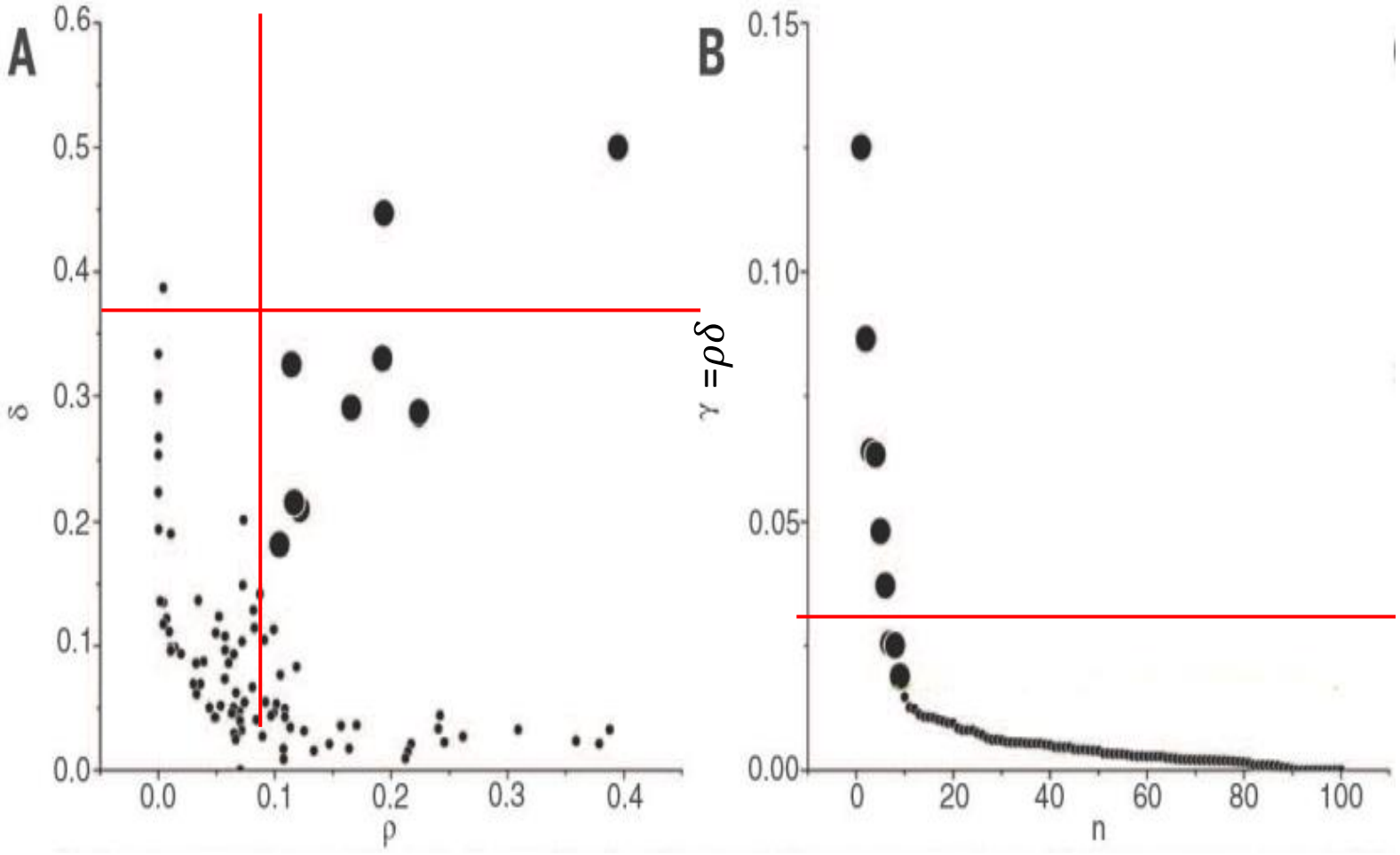


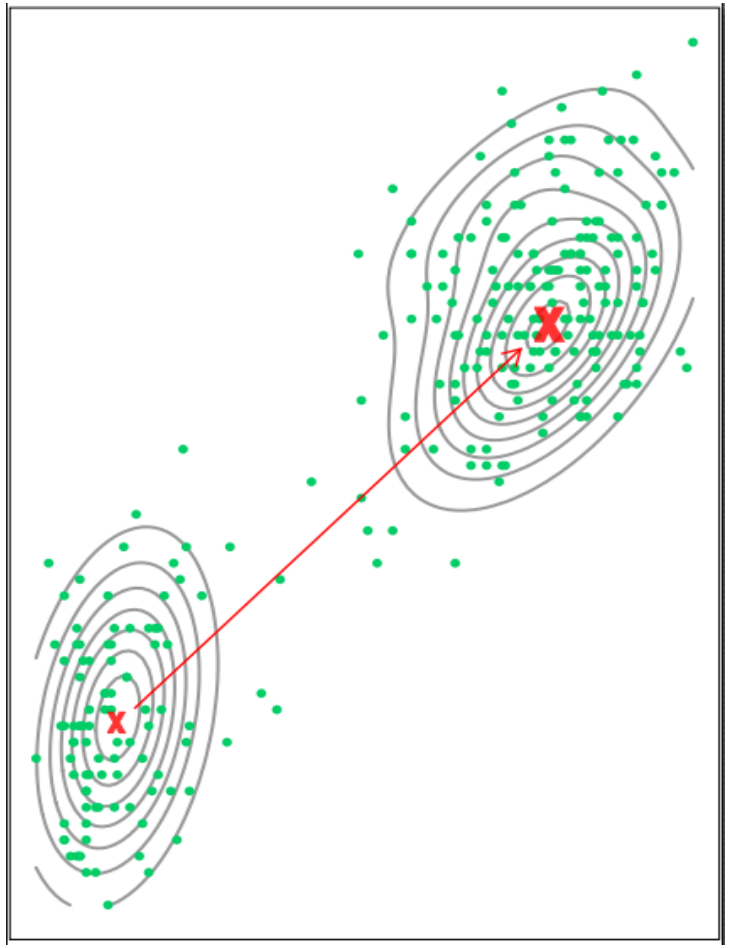
Part 1 – Peak Finding



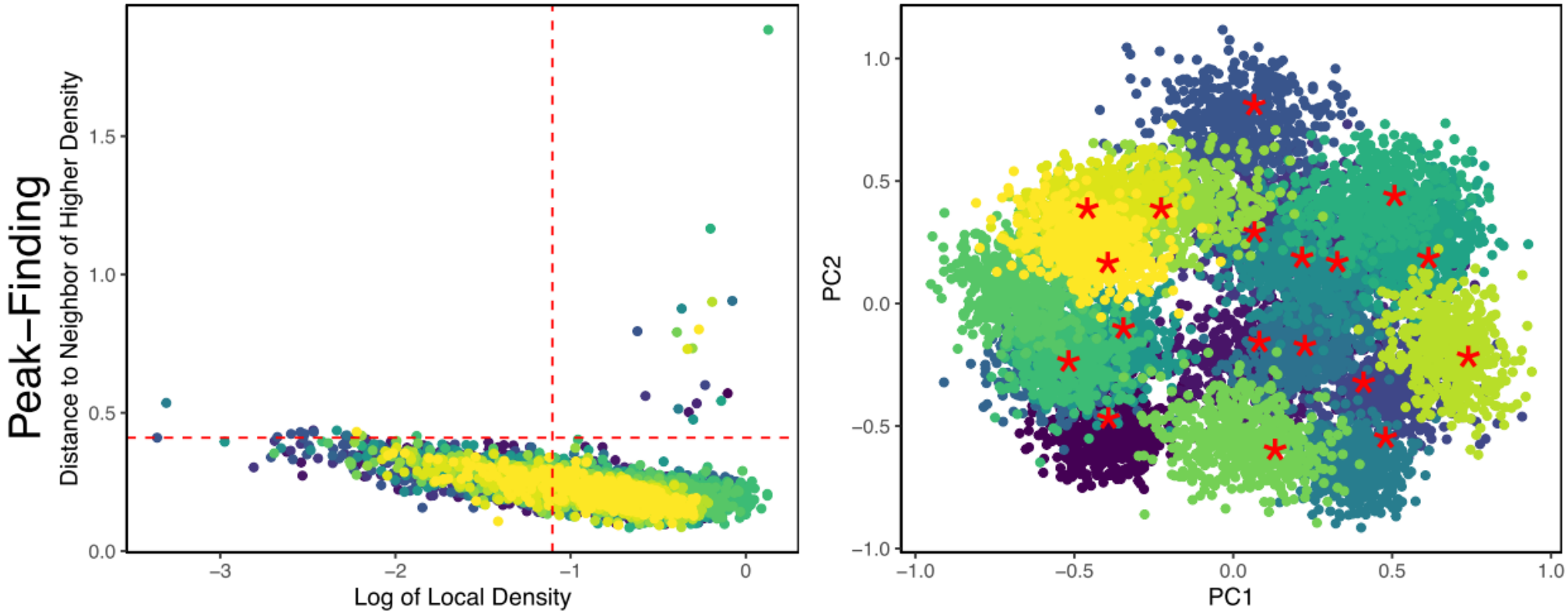


Part 1 – Peak Finding



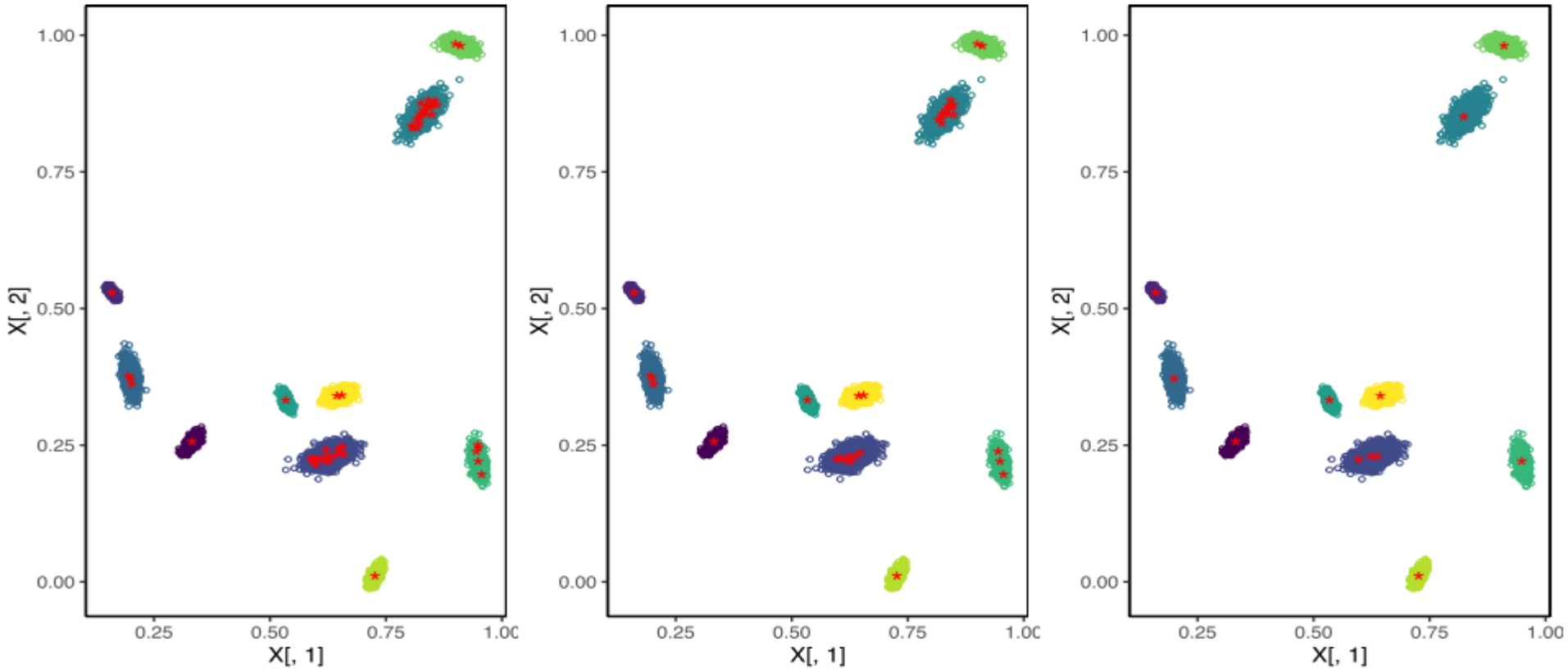


Part 1 – Peak Finding



A 10-dimensional dataset contains 20 components. Clusters are indicated by different colours. The right figure shows the locations of the selected peaks, projected onto the first two principal components.

Part 1 – Peak Finding



Exemplars selected for different cut-off levels on the density $\rho(\boldsymbol{x})$ and distance $\delta(\boldsymbol{x})$. Left: 10^{th}-percentile of $\rho(\boldsymbol{x})$ & 97.5^{th}-p of $\delta(\boldsymbol{x})$. Center: 20^{th}-p of $\rho(\boldsymbol{x})$ & 97.5^{th}-p of $\delta(\boldsymbol{x})$. Right: 20^{th}-p of $\rho(\boldsymbol{x})$ & 99.5^{th}-p of $\delta(\boldsymbol{x})$.

Part 1 – Peak Finding

Theorem: For n large enough, with high probability, $\mathcal{E}_0$ contains unique estimates for all the true modes of the GMM.

Part 1 – Peak Finding

Theorem: For n large enough, with high probability, $\mathcal{E}_0$ contains unique estimates for all the true modes of the GMM.

Relaxing the cut-off levels on the density $\rho(\boldsymbol{x})$ and distance $\delta(\boldsymbol{x})$, $\mathcal{E}_0$ is guaranteed to include all modes in the data.

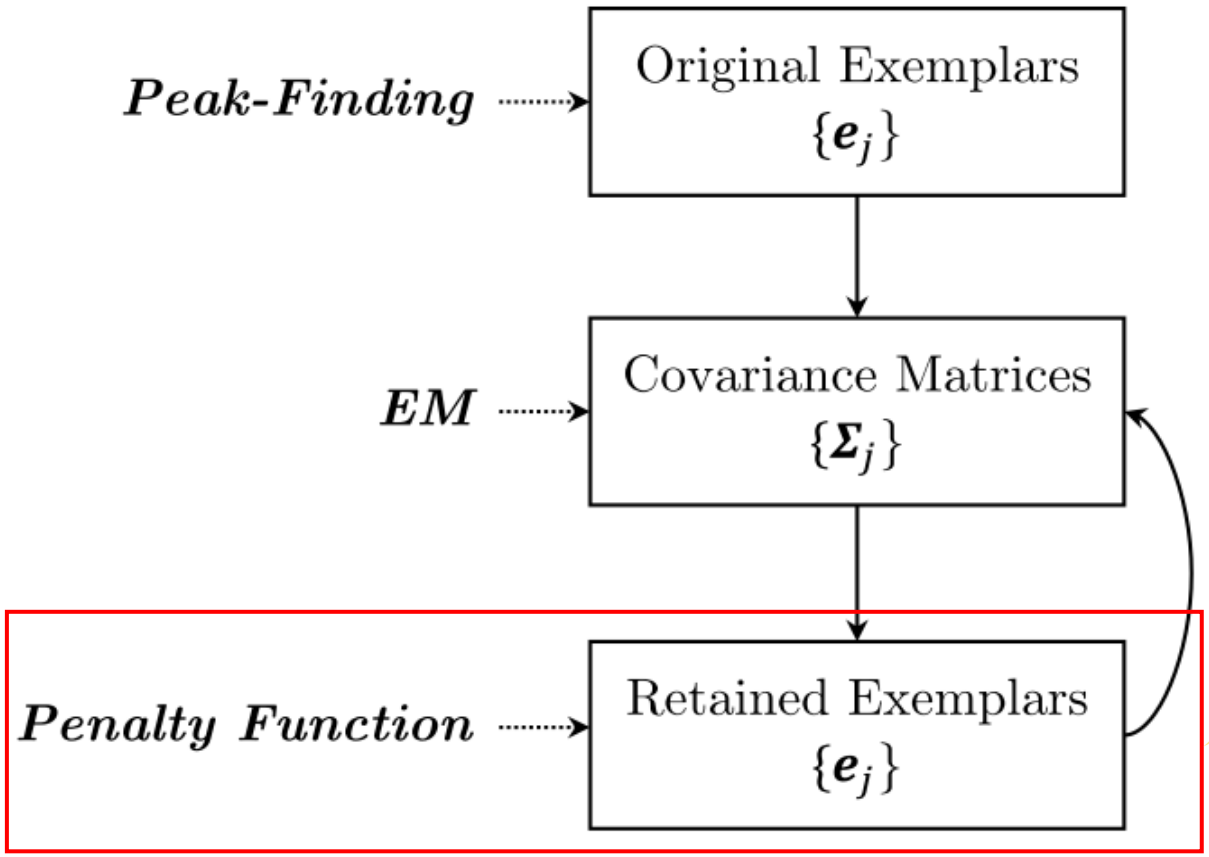
Part 1 – Peak Finding

Theorem: For n large enough, with high probability, $\mathcal{E}_0$ contains unique estimates for all the true modes of the GMM.

Relaxing the cut-off levels on the density $\rho(\boldsymbol{x})$ and distance $\delta(\boldsymbol{x})$, $\mathcal{E}_0$ is guaranteed to include all modes in the data.

Apply a pruning strategy to retain only instances that well represent their associated Gaussian components.

Part 2 – Exemplar Pruning



When prune the exemplars $\{e_j\}_{j=1}^{\kappa}$, the covariance estimates $\{\Sigma_j\}_{j=1}^{\kappa}$ are fixed.

Part 2 – Exemplar Pruning

Given $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, the log-likelihood function is

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right).$$

Introduce sparsity into $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_\kappa)$; if $\pi_j = 0$, then the exemplar $\boldsymbol{e}_j$ is dismissed as cluster centre.

Penalty function:

log-likelihood + cardinality penalty of $\boldsymbol{\pi}$

Part 2 – Exemplar Pruning

Given $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, the log-likelihood function is

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right).$$

Introduce sparsity into $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_\kappa)$; if $\pi_j = 0$, then the exemplar $\boldsymbol{e}_j$ is dismissed as cluster centre.

Penalty function:

log-likelihood + cardinality penalty of $\boldsymbol{\pi}$

Part 2 – Exemplar Pruning

Given $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, the log-likelihood function is

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right).$$

By Jensen's inequality we have

$$-\log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right) \leq \sum_{j=1}^{\kappa} r_{ij} \log\left(\frac{r_{ij}}{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)}\right).$$

r_{ij}'s are the responsibilities in the EM algorithm: $\pi_j = \frac{1}{n}\sum_{i=1}^{n} r_{ij}$ and $\sum_{j=1}^{\kappa} r_{ij}$=1.

Part 2 – Exemplar Pruning

Given $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, the log-likelihood function is

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right).$$

Therefore, the negative log-likelihood is

$$-\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right) = \min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij} \log\left(\frac{r_{ij}}{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)}\right).$$

Part 2 – Exemplar Pruning

Given $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, the log-likelihood function is

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right).$$

Minimizing the negative log-likelihood is equivalent to

$$\min_{\{\boldsymbol{\Sigma}_j \succ 0\}_{j=1}^{\kappa}} \min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij} \log\left(\frac{r_{ij}}{\pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)}\right),$$

which is

$$\min_{\{\boldsymbol{\Sigma}_j \succ 0\}_{j=1}^{\kappa}} \min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij} \left[\log\left(\frac{r_{ij}}{\pi_j}\right) + \frac{1}{2}\log(|\Sigma_j|) + \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{e}_j)'\Sigma_j^{-1}(\boldsymbol{x}_i - \boldsymbol{e}_j)\right]$$

Part 2 – Exemplar Pruning

$$\min_{\{\boldsymbol{\Sigma}_j \succ 0\}_{j=1}^{\kappa}} \min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\left[\log\left(\frac{r_{ij}}{\pi_j}\right) + \frac{1}{2}\log\left(\left|\Sigma_j\right|\right) + \frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{e}_j\right)'\Sigma_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{e}_j\right)\right]$$

- For exemplar pruning, the covariance estimates are fixed.
- Take $\log\left(\frac{r_{ij}}{\pi_j}\right)$ out. Otherwise, numerical algorithms will behave erratically for any $\pi_j \to 0$.

We obtain

$$\min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\left[\frac{1}{2}\log\left(\left|\Sigma_j\right|\right) + \frac{1}{2}\left(\boldsymbol{x}_i - \boldsymbol{e}_j\right)'\Sigma_j^{-1}\left(\boldsymbol{x}_i - \boldsymbol{e}_j\right)\right].$$

Part 2 – Exemplar Pruning

Given $\mathcal{E}_0 = \{\boldsymbol{e}_j\}_{j=1}^{\kappa}$, the log-likelihood function is

$$\sum_{i=1}^{n} \log\left(\sum_{j=1}^{\kappa} \pi_j \phi(\boldsymbol{x}_i; \boldsymbol{e}_j, \boldsymbol{\Sigma}_j)\right).$$

Introduce sparsity into $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_\kappa)$; if $\pi_j = 0$, then the exemplar $\boldsymbol{e}_j$ is dismissed as cluster centre.

Penalty function:

log-likelihood + cardinality penalty of $\boldsymbol{\pi}$

Part 2 – Exemplar Pruning

Most learning methods with sparsity constraints apply convex relaxations, leading to optimization with the L1-norm.

The classical L1-norm penalty is not suitable here: $\| \boldsymbol{\pi} \|_1 = 1$ is constant on the simplex.

Our penalty is in the form of $\| \boldsymbol{\delta} \circ \boldsymbol{\pi} \|_1$, where $\circ$ is the element-wise multiplication operator.

The weight vector $\boldsymbol{\delta}$ should be data-driven and has the desirable property that gives more penalty to closer exemplars.

Part 2 – Exemplar Pruning

The weight vector $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_\kappa)$ is computed as

$$\delta_i = \max_{j=1,\ldots,\kappa} \left\{ \Phi\left(-\frac{1}{2} \sqrt{\left(\boldsymbol{e}_i - \boldsymbol{e}_j\right)' \Sigma_j^{-1} \left(\boldsymbol{e}_i - \boldsymbol{e}_j\right)} \right) \right\}.$$

The Mahalanobis distance is measured by the covariance matrix of the target exemplar, not the covariance matrix of $\boldsymbol{e}_i$.

This is because the exemplar $\boldsymbol{e}_i$, if pruned, should be assigned into the cluster of its nearest exemplar, where the proximity will be measured by the covariance matrix of the destination exemplar.

Part 2 – Exemplar Pruning

The weight vector $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_\kappa)$ is computed as

$$\delta_i = \max_{j=1,\ldots,\kappa} \left\{ \Phi\left(-\frac{1}{2} \sqrt{(\boldsymbol{e}_i - \boldsymbol{e}_j)' \Sigma_j^{-1} (\boldsymbol{e}_i - \boldsymbol{e}_j)} \right) \right\}.$$

Interpretation: the weight δ_i

(1) reflects the likelihood of the exemplar $\boldsymbol{e}_i$ belonging to the group of another exemplar,

(2) measures the overlapping degree between the component distributions of $\boldsymbol{e}_i$ and $\boldsymbol{e}_j$, assuming common prior probability and covariance.

Part 2 – Exemplar Pruning

Penalty function:

log-likelihood + cardinality penalty of $\boldsymbol{\pi}$

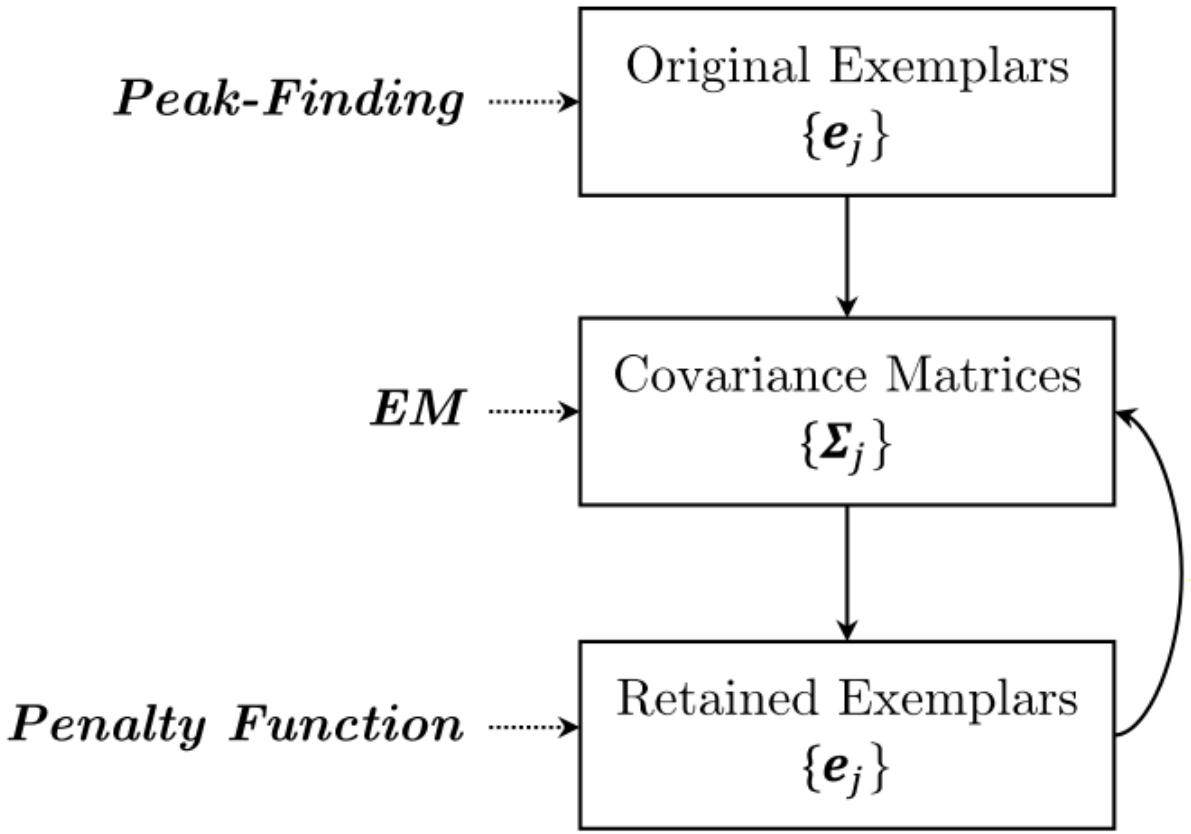
$$\min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\sum_{j=1}^{\kappa} r_{ij}\left[\frac{1}{2}\log(|\Sigma_j|) + \frac{1}{2}(\boldsymbol{x}_i - \boldsymbol{e}_j)'\Sigma_j^{-1}(\boldsymbol{x}_i - \boldsymbol{e}_j)\right] + \theta\sum_{i=1}^{n}\boldsymbol{r}_i^T\boldsymbol{\delta}$$

Equivalent to

$$\min_{\{\boldsymbol{r}_i \in \triangle\}_{i=1}^{n}} \sum_{i=1}^{n}\boldsymbol{r}_i^T\left(\frac{1}{2}\boldsymbol{\xi} + \frac{1}{2}\boldsymbol{d}_i + \theta\boldsymbol{\delta}\right),$$

extremely simple separable convex function.

Part 2 – Exemplar Pruning



Update the covariance estimates $\{\Sigma_j\}_{j=1}^{\kappa}$ via the EM algorithm.

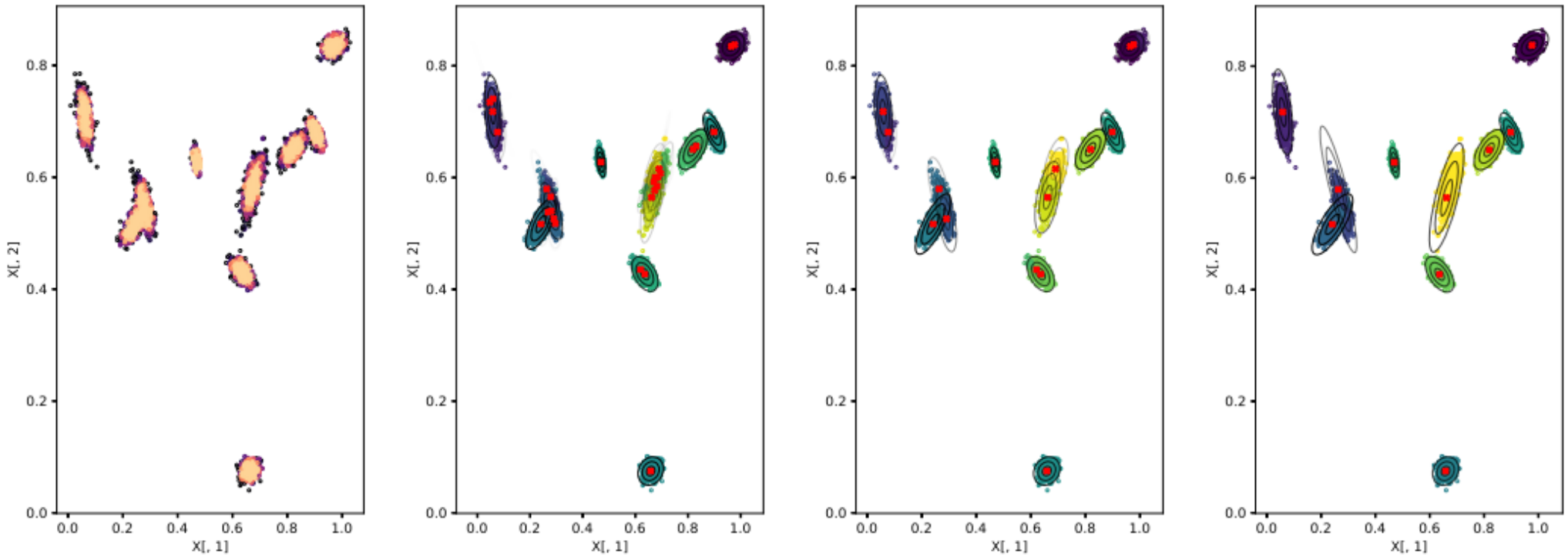


Fig. 4: A worked example of the clustering process for a 2-dimensional dataset with 10 components. (1) The leftmost figure shows the kernel density estimate for each instance, with lighter colors representing instances of higher density. (2) The second figure shows the initial exemplars (in red) with confidence ellipses representing the initial covariance matrix for each exemplar. (3) The third figure shows an intermediate clustering step, when multiple exemplars have been pruned from the initial set. (4) The rightmost figure shows the optimal clustering selected from the sequence using the ICL criterion.

Evaluation

Synthetic data were generated with the following configurations:

– Number of observations: $n \in \{ 10000 \}$.
– Dimension of the data: $p \in \{ 2, 10, 20 \}$.
– Number of mixture components: $m \in \{ 10, 40 \}$.
– Overlapping degree between components: $o \in \{ 0, 0.1 \}$.

For each of the 12 simulation configurations, ten datasets were generated. The results reported are averaged over the ten datasets for each configuration.

Evaluation

Real-world datasets are

Name	Instances	Dim	Classes
Dermatology	358	34	6
Ecoli	336	7	8
Optdigits	5620	64	10
Pendigits	10992	16	10
Seeds	210	7	3

Evaluation

Benchmark methods are

– Mclust (McL). This approach uses model-based hierarchical agglomerative clustering to provide initial partitions for the EM algorithm.

– MixMod emEM (MmE). This approach uses several short runs of EM with random initialization to quickly decide the best model for the full run of EM.

– MixMod Random (MmR). This approach uses several random initializations for the EM algorithm.

Evaluation

Data			Ours		McL		MmE		MmR	
d	m	o_{ij}	ARI	Time (s)	ARI	Time (s)	ARI	Time (s)	ARI	Time (s)
2	10	0	**1.00**	32.03	0.96	**3.20**	0.96	1611.54	0.94	1208.54
		0.1	**0.94**	42.92	0.93	**9.87**	0.93	1805.38	0.90	1442.59
	40	0	**1.00**	**39.32**	**1.00**	59.20	0.92	54189.47	0.96	43486.47
		0.1	**0.98**	64.14	0.70	**12.99**	-	-	-	-
10	10	0	**1.00**	**41.48**	**1.00**	57.10	0.99	5699.92	0.99	3791.07
		0.1	0.81	**201.31**	**0.90**	229.3	0.90	3036.65	0.90	3925.71
	40	0	**1.00**	**55.34**	**1.00**	59.20	-	-	-	-
		0.1	0.63	**35.20**	**0.75**	619.43	-	-	-	-
20	10	0	**1.00**	**83.65**	0.98	158.73	1.00	8822.95	0.98	6886.02
		0.1	0.62	**132.67**	0.62	230.08	**0.66**	10643.29	0.58	7892.44
	40	0	**1.00**	89.47	0.98	**73.28**	-	-	-	-
		0.1	**0.14**	**928.36**	0.03	1667.0.2	-	-	-	-

Table 2: The quality of the clusterings and execution time on the simulated datasets. Values in the table are averages computed from the ten simulated datasets for each configuration. The best results are highlighted in bold.

Evaluation

Data Name	Ours ARI	Ours Time (s)	McL ARI	McL Time (s)	MmE ARI	MmE Time (s)	MmR ARI	MmR Time (s)
Dermatology	**0.62**	93.98	0.20	**0.47**	0.00	2632.38	0.00	1813.19
Ecoli	0.59	38.02	0.21	**1.08**	**0.62**	102.32	0.60	83.60
Optdigits	**0.52**	**110.69**	0.43	129.96	-	-	-	-
Pendigits	**0.69**	**50.71**	0.57	95.77	-	-	-	-
Seeds	0.75	0.27	**0.79**	**0.08**	0.56	17.73	0.36	20.30

Table 3: The quality of the clusterings and execution time on the real-world datasets. The best results are highlighted in bold.

Evaluation

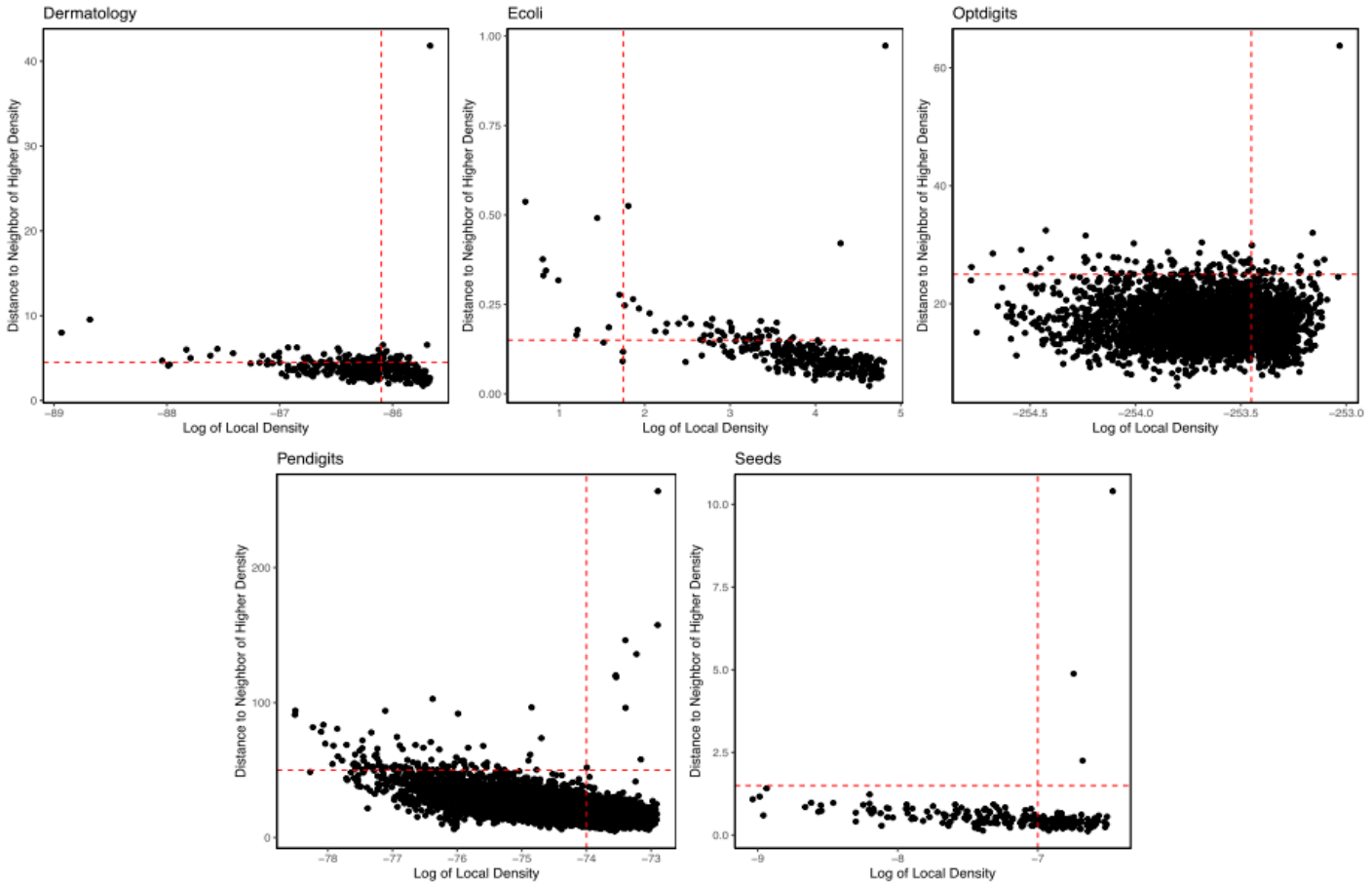


Fig. 5: The plot of the local density against the distance to nearest neighbor of higher local density for the real-world datasets. The thresholds are set so as to include all instances that could be reasonably considered as potential exemplars.

Evaluation

Data			l= 10th				l= 20th			
			τ= 97th		τ= 98th		τ= 97th		τ= 98th	
d	m	o_{ij}	ARI	Time (s)	ARI	Time (s)	ARI	Time (s)	ARI	Time (s)
10	10	0	**1.00**	63.08	**1.00**	27.13	**1.00**	46.93	**1.00**	**23.20**
		0.1	**0.80**	1336.15	0.78	385.91	0.77	486.93	0.76	**91.86**
	40	0	**1.00**	286.19	**1.00**	27.30	**1.00**	50.52	**1.00**	**20.20**
		0.1	**0.85**	2783.22	0.83	793.87	0.82	1365.00	0.80	**312.12**

Table 4: Comparison of clustering quality (ARI) and execution time (s) for combinations of the threshold parameter values l and τ: $l \in \{10\text{th}, 20\text{th}\}$-percentile of $f_h(\boldsymbol{x})$ values, and $\tau \in \{97\text{th}, 98\text{th}\}$-percentile of $\omega(\boldsymbol{x})$ values.

Reference

- Tobin, J., Ho, C.P., Zhang, M.: An Efficient Algorithm for Model-Based Clustering using Exemplar Means and L1 Regularization. Submitted to "European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD 2022)".
- [**Exemplar**] Lashkari, D., Golland, P.: Convex clustering with exemplar-based models. In: Advances in Neural Information Processing Systems 20 (NIPS 2007), pp. 825–832 (2008)
- [**Exemplar**] Pilanci, M., Ghaoui, L.E., Chandrasekaran, V.: Recovery of sparse probability measures via convex programming. In: Advances in Neural Information Processing Systems 25 (NIPS 2012), pp. 2420–2428 (2012)
- [**Peak Finding**] Rodriguez, A., Laio, A.: Clustering by fast search and find of density peaks. Science (New York, N.Y.) 344(6191), 1492–1496 (2014)