Behavioural pattern recognition of animal paths obtained from experimental procedures

Avgoustinos Vouros[1]

[1]PhD student,
Department of Computer Science,
University of Sheffield

Supervised by Prof Eleni Vasilaki

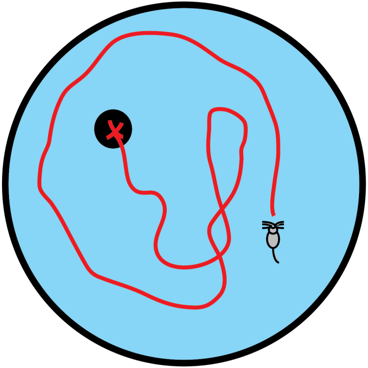
Behavioural analysis inside the Morris Water Maze

The Morris Water Maze (MWM)

It was designed by Richard Morris in 1981.

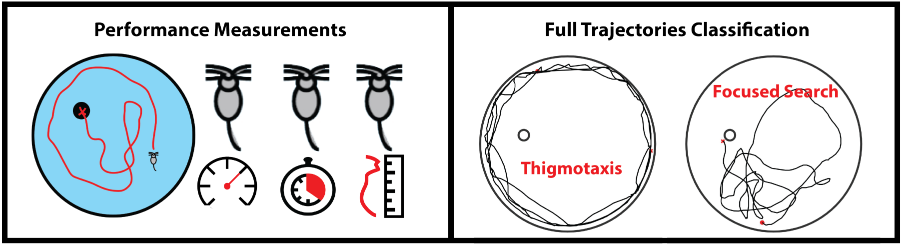
It is one of the most widely used tasks in behavioural neuroscience. More than 2000 publications within the decade 1990-2001 [1].

It is used to study the psychological processes and neural mechanisms of spatial learning and memory.

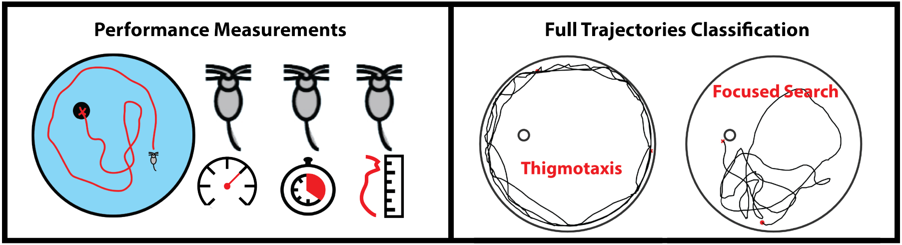


[1] D'Hooge, Rudi, and Peter P. De Deyn. "Applications of the Morris water maze in the study of learning and memory." Brain research reviews 36.1 (2001): 60-90.

Data Analysis in the MWM



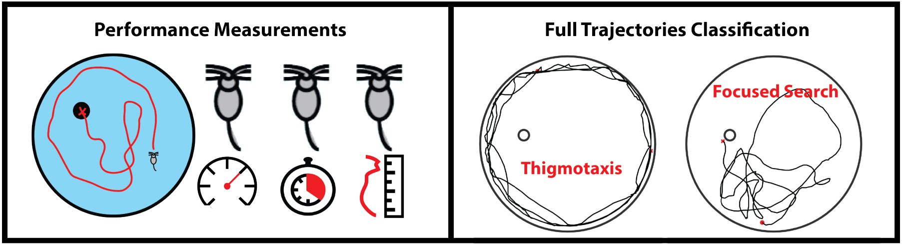
Data Analysis in the MWM



Performance measurements: Insufficient to capture all the different animal behaviours that are present during the experiments [1].

[1] Dalm, Sergiu, et al. "Quantification of swim patterns in the Morris water maze." Behavior Research Methods, Instruments, & Computers 32.1 (2000): 134-139.

Data Analysis in the MWM



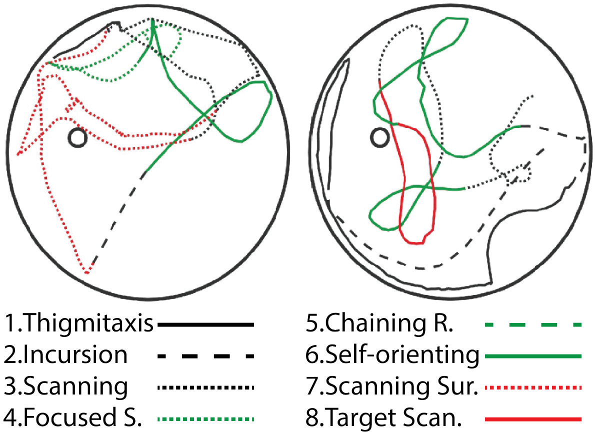
Performance measurements: Insufficient to capture all the different animal behaviours that are present during the experiments [1].

Full trajectories classification: Animals employ several behaviours during each trial in order to find the platform and by assigning whole animal trajectories to single behavioural classes results in the loss of important information [2].

[1] Dalm, Sergiu, et al. "Quantification of swim patterns in the Morris water maze." Behavior Research Methods, Instruments, & Computers 32.1 (2000): 134-139.
[2] Gehring, Tiago V., et al. "Detailed classification of swimming paths in the Morris Water Maze: multiple strategies within one trial." Scientific reports 5 (2015): 14562.

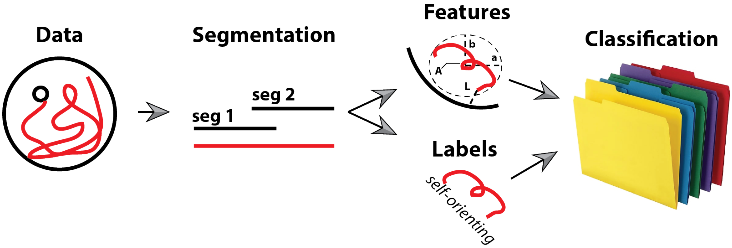
Data Analysis in the MWM

Detailed Trajectories Classification

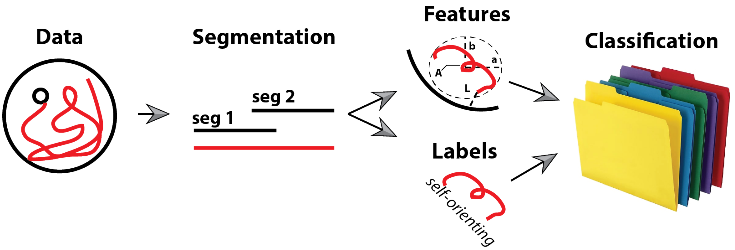


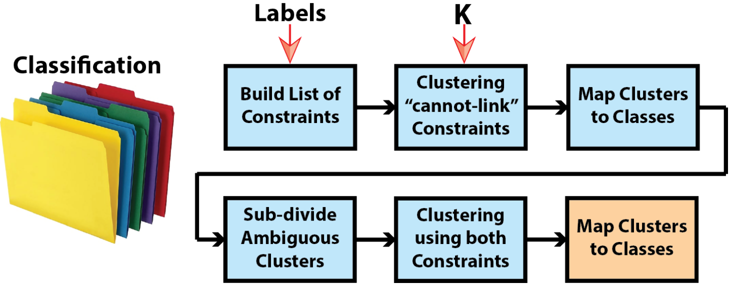
[1] Gehring, Tiago V., et al. "Detailed classification of swimming paths in the Morris Water Maze: multiple strategies within one trial." Scientific reports 5 (2015): 14562.

Procedure of Gehring et al.

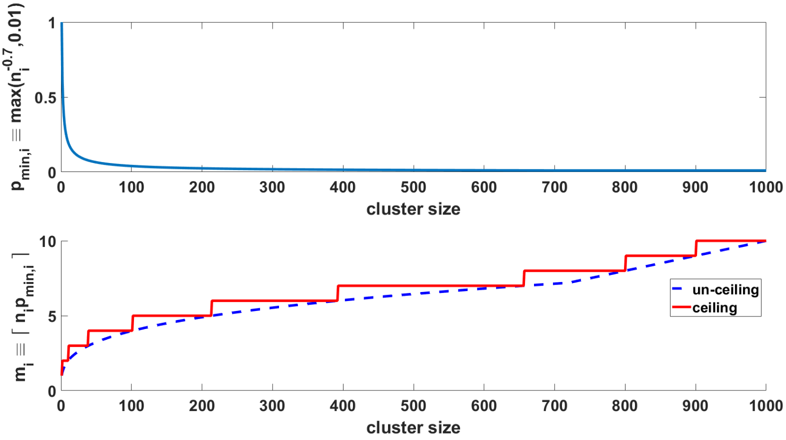


Procedure of Gehring et al.



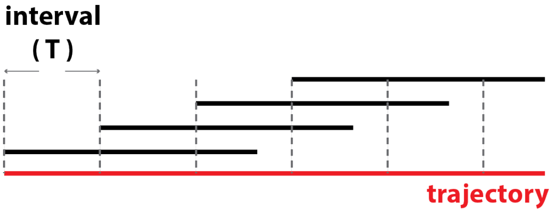


Mapping clusters to classes

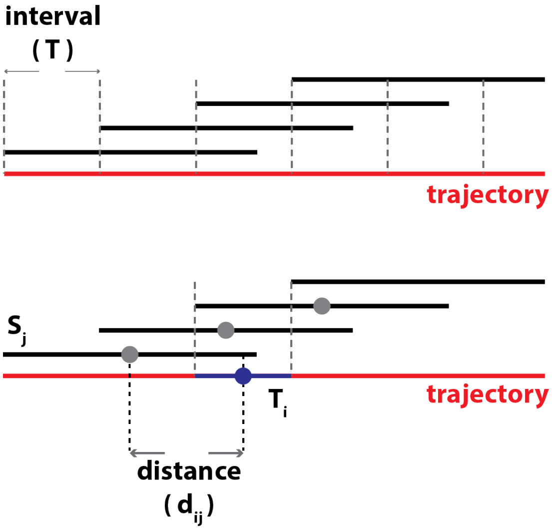


Procedure of Gehring et al.

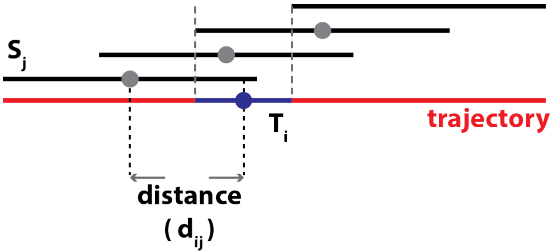
Mapping segments back to the original trajectories



Mapping segments back to the original trajectories

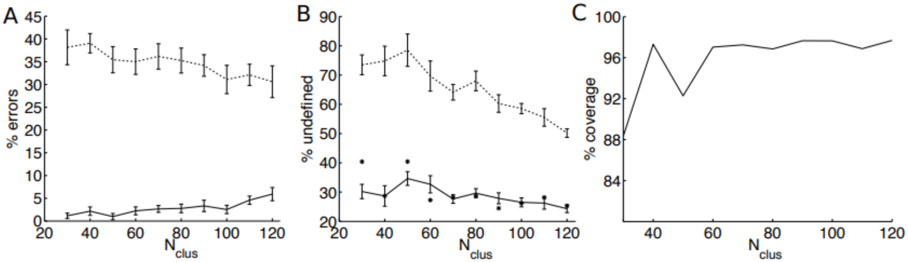


Mapping segments back to the original trajectories



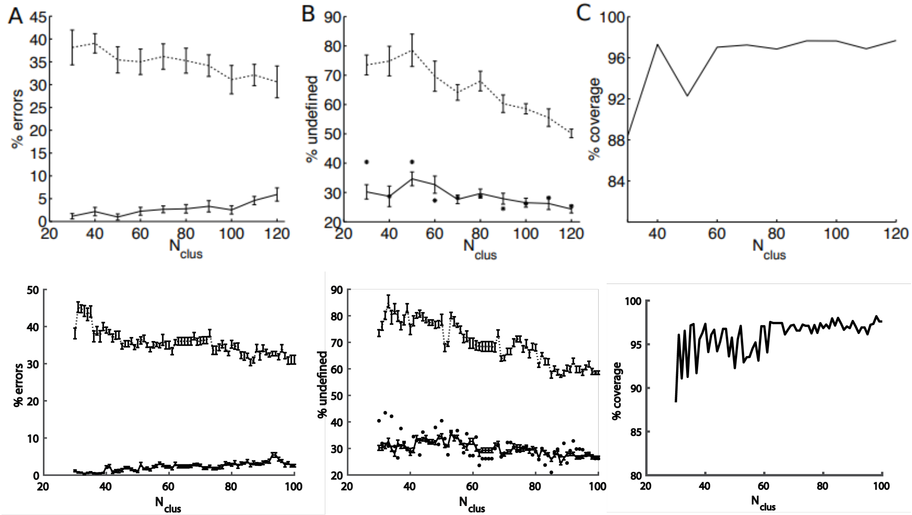
$$C_{T_i} \equiv arg_{c_k} max \sum_{\left(\substack{S_j \in c_k \\ T_i \cap S_j \neq \varnothing}\right)} w_k \cdot e^{-\frac{d_{ij}^2}{2 \cdot \sigma^2}}, \qquad \sigma = 4, \quad w_k = \frac{L_{max}^{cont}}{L_{max,k}}$$

How to find K?


A
45
40
35
30
25
20
15
10
5
0
% errors
20
40
60
80
100
120
N clus
B
90
80
70
60
50
40
30
20
% undefined
20
40
60
80
100
120
N clus
C
100
96
92
88
84
% coverage
40
60
80
100
120
N clus

Procedure of Gehring et al.

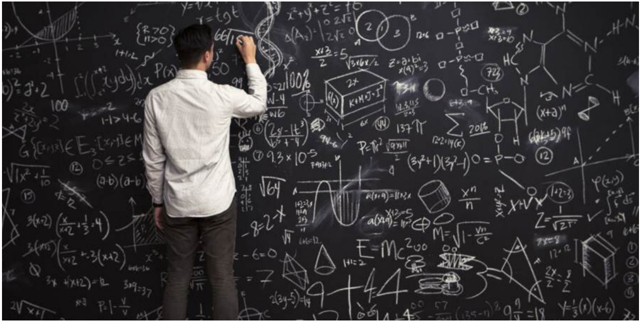
How to find K?



Procedure of Gehring et al.

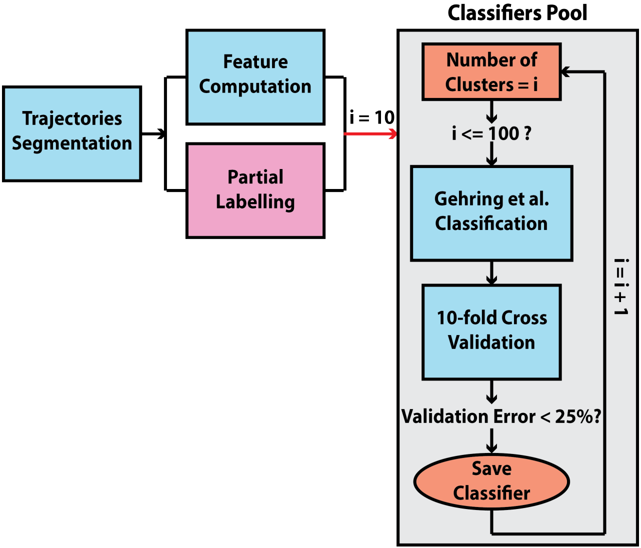
- Segmentation tuning.
- Labelling.
- Classification tuning.
- Final conclusions are based on different segmentation tunings combined together.

How to find K?

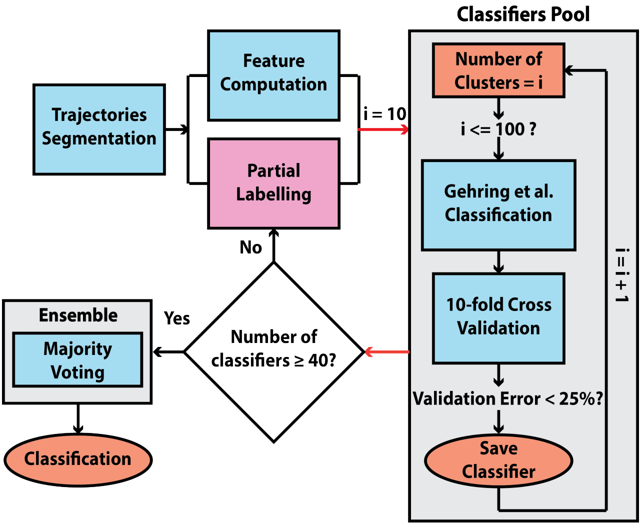


Procedure of Vouros et al.

Classification boosting with majority voting

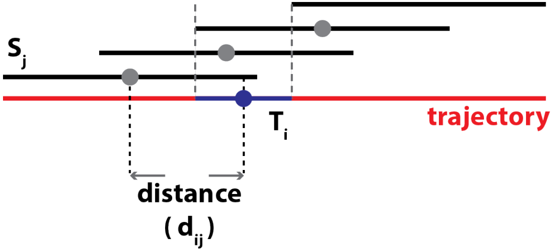


Classification boosting with majority voting



Procedure of Vouros et al.

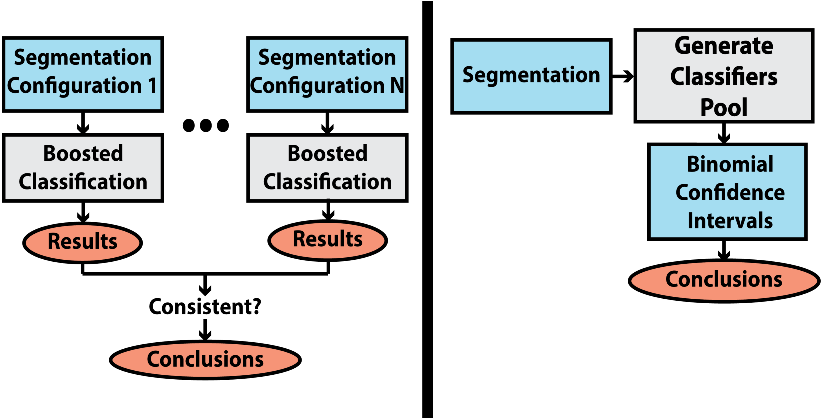
Mapping segments back to the original trajectories:
segmentation independent, T and σ proportional to R



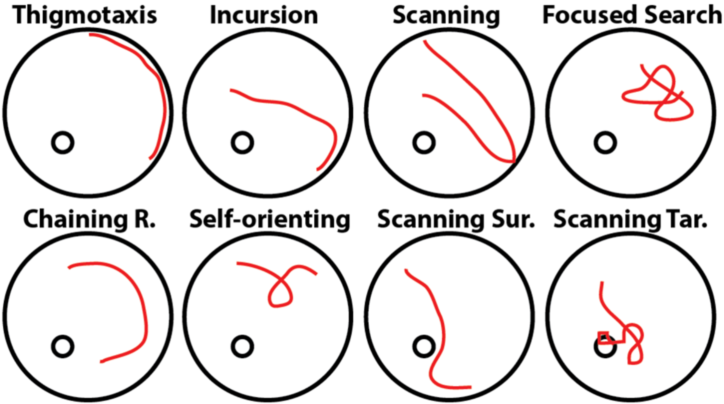
$$C_{T_i} \equiv arg_{c_k} max \sum_{\binom{S_j \in c_k}{T_i \cap S_j \neq \varnothing}} w_k \cdot e^{-\frac{d_{ij}^2}{2 \cdot \sigma^2}}, \qquad w_k = \frac{1}{P(c_k)}$$

Procedure of Vouros et al.

Validation and Confidence

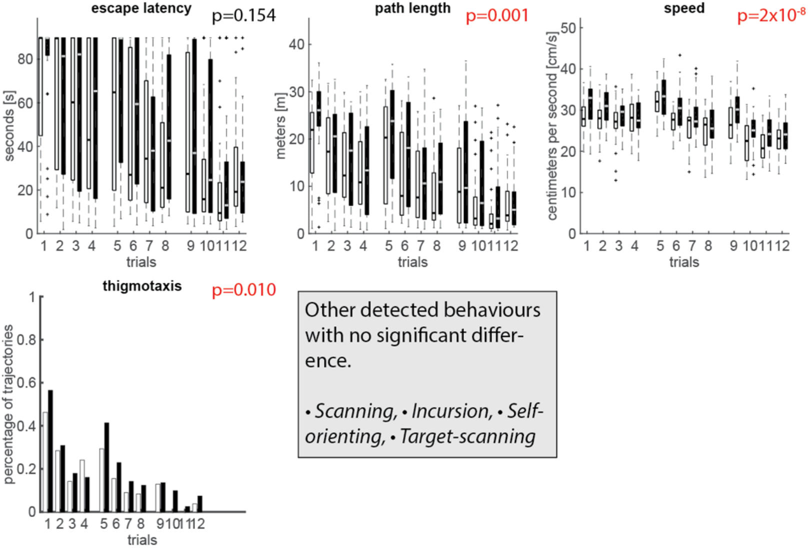


Results


Thigmotaxis
Incursion
Scanning
Focused Search
Chaining R.
Self-orienting
Scanning Sur.
Scanning Tar.

Results: EPFL - Stress vs Control Groups

Performance Measurements & Full Trajectory Analysis

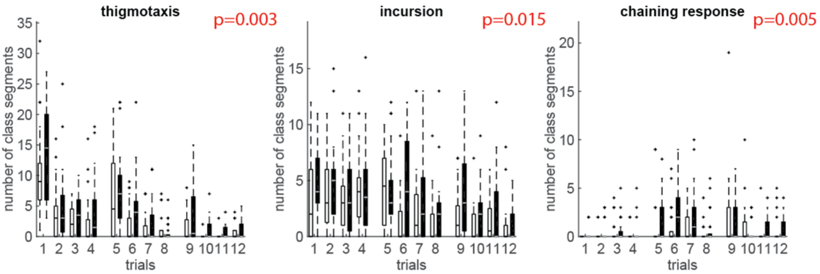


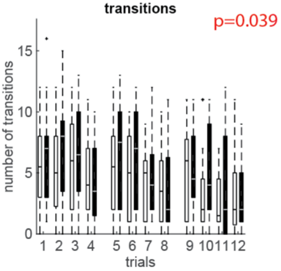
Other detected behaviours with no significant difference.

• *Scanning*, • *Incursion*, • *Self-orienting*, • *Target-scanning*

Results: EPFL - Stress vs Control Groups

Our Method of: Trajectory Segmentation Analysis





Other detected behaviours with no significant difference.

- *Scanning*, • *Focused-search*, • *Self-orienting*, • *Scanning-surroundings*, • *Target-scanning*

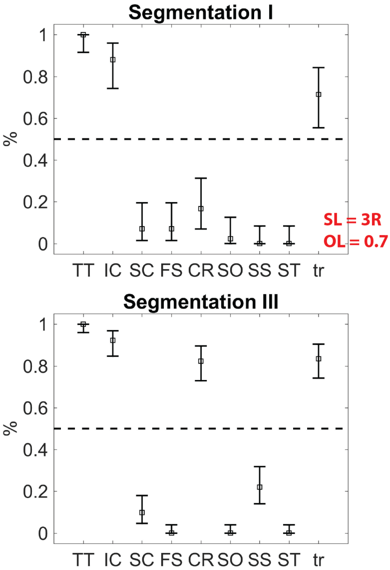
CONCLUSION

Stress animal group (black) finds the platform as fast as the control group even though it runs faster and swaps longer paths **because it implements low level strategies more often.**

Results: EPFL - Stress vs Control Groups

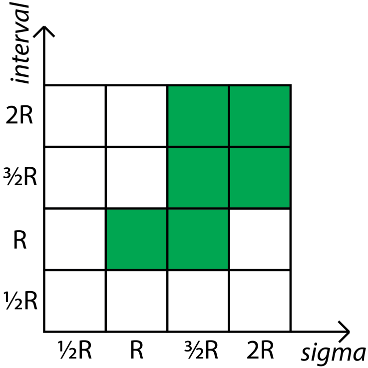
Ensemble Result (Friedman test p-values per strategy and transitions, a = 0.05)									
Segmentation	**TT**	**IC**	**SC**	**FS**	**CR**	**SO**	**SS**	**ST**	**tr**
3R, 0.7	**0.008**	**0.011**	0.450	0.205	0.156	0.960	0.271	0.571	**0.035**
2.5R, 0.7	**0.005**	**0.013**	0.157	0.278	**0.003**	0.638	0.190	0.345	**0.019**
2.5R, 0.9	**0.004**	**0.009**	0.501	0.444	**0.007**	0.718	0.229	0.827	**0.037**
2R, 0.7	**0.004**	**0.005**	0.156	0.821	**0.008**	0.749	0.436	0.389	**0.038**

Results: EPFL - Stress vs Control Groups



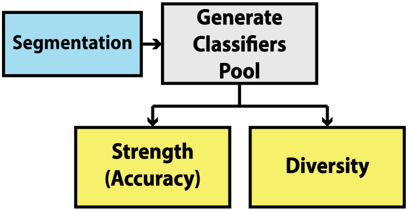
What about interval length and σ?

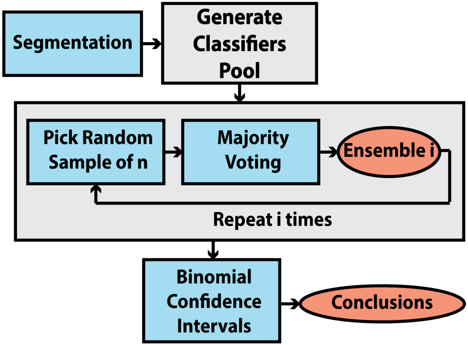
$$C_{T_i} \equiv arg_{c_k} max \sum_{\binom{S_j \in c_k}{T_i \cap S_j \neq \varnothing}} w_k \cdot e^{-\frac{d_{ij}^2}{2\cdot\sigma^2}}, \qquad w_k = \frac{1}{P(c_k)}$$



Further validation: EPFL - Stress vs Control Groups

Diversity [1-2], and strength [3-4] of the classifiers?



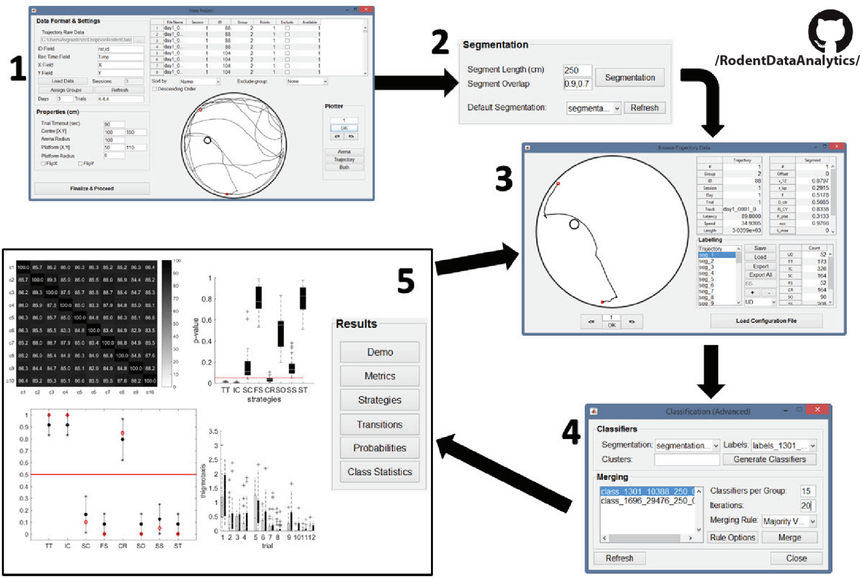


[1] Gerecke, Uwe, Noel E. Sharkey, and Amanda JC Sharkey. "Common evidence vectors for self-organized ensemble localization." Neurocomputing 55.3-4 (2003): 499-519.
[2] Schapire, Robert E. "The strength of weak learnability." Machine learning 5.2 (1990): 197-227.
[3] Zhu, Mu. "Use of majority votes in statistical learning." Wiley Interdisciplinary Reviews: Computational Statistics 7.6 (2015): 357-371.
[4] Ruta, Dymitr, and Bogdan Gabrys. "A theoretical analysis of the limits of majority voting errors for multiple classifier systems." Pattern Analysis and Applications 5.4 (2002): 333-350.

Diversity, and strength of the classifiers?

	Segmentation I	Segmentation II	Segmentation III	Segmentation IV
Number of generated Classifiers	42	78	91	64
	Performance: Classifiers			
Average Error (%) [min-max]	16.8 [5.4 24.9]	17.5 [3.7 25.0]	13.9 [1.8 21.5]	18.0 [7.3 24.9]
Unclassified (%) Segments	2.5	2.5	1.3	3.7
Agreement (%)	58.7	61.0	59.6	56.3
	Performance: Ensemble(s)			
Error (%)	0.0	0.2	0.0	0.0
Unclassified (%) Segments	0.0	0.0	0.0	0.1
Agreement (%)	83.4	82.6	82.3	80.0

The RODA Software



Further applications, requests and Q&A

nencki institute of experimental biology

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

- Niina Lapinlampi, University of Eastern Finland, A.I. Virtanen Institute for Molecular Sciences, Finland.
- Gido Gravesteijn, CADASIL research group, Leiden University Medical Center, Department of Clinical Genetics and Department of Human Genetics, Leiden, The Netherlands.
- Richard Pinnell and Ulrich Hofmann Neuroelectronic Systems, Dept. of Neurosurgery, University Medical Centre Freiburg, Freiburg, Germany.
- Qazi Rahman, King's College London, Psychology Department, Health Psychology Research Group, UK.
- Noam Joseph, Mote Marine Laboratory & Aquarium, Florida, USA (now in Israel).

Manuscript under peer-review by Scientific Reports

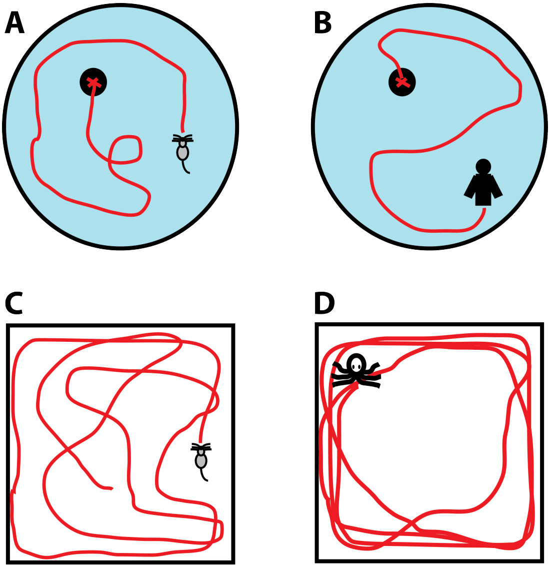
A generalised framework for detailed classification of swimming paths inside the Morris Water Maze

Avgoustinos Vouros[1,*], Tiago V. Gehring[1], Kinga Szydlowska[2], Artur Janusz[2], Zehai Tu[1], Mike Croucher[1], Katarzyna Lukasiuk[2], Witold Konopka[2], Carmen Sandi[3], and Eleni Vasilaki[1,+]

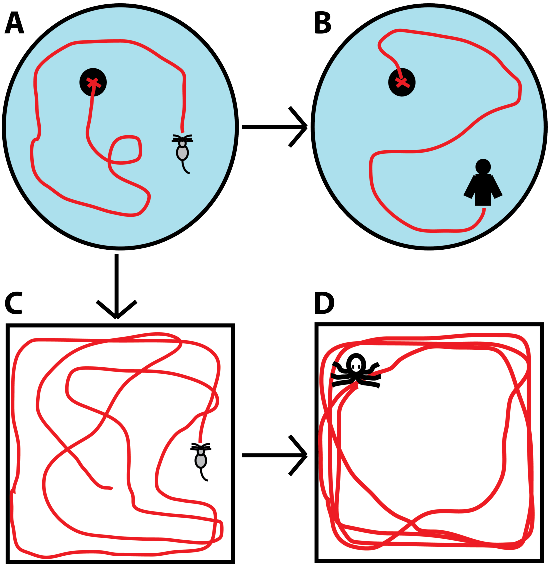
[1]Department of Computer Science, The University of Sheffield, Sheffield, UK
[2]Department of Molecular and Cellular Neurobiology, Nencki Institute of Experimental Biology, Warsaw, Poland
[3]Laboratory of Behavioral Genetics, Brain Mind Institute, EPFL, Lausanne, Switzerland
[*]avouros1@sheffield.ac.uk
[+]e.vasilaki@sheffield.ac.uk

RODA adaptation to other experimental procedures

RODA adaptation to other experimental procedures



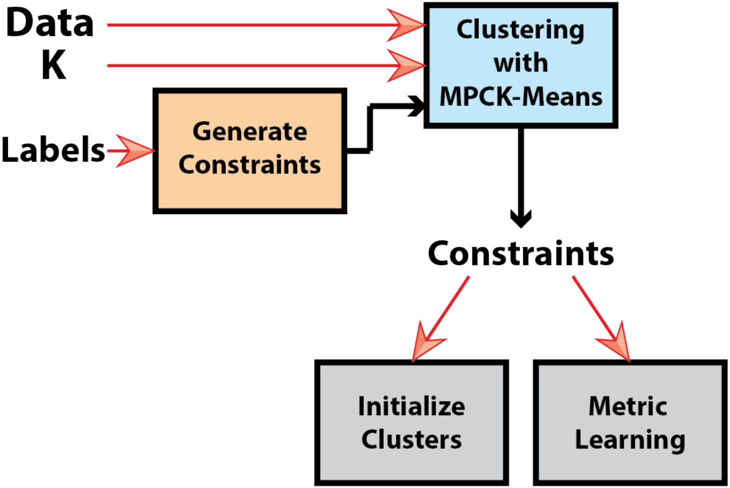
RODA adaptation to other experimental procedures



RODA adaptation to other experimental procedures

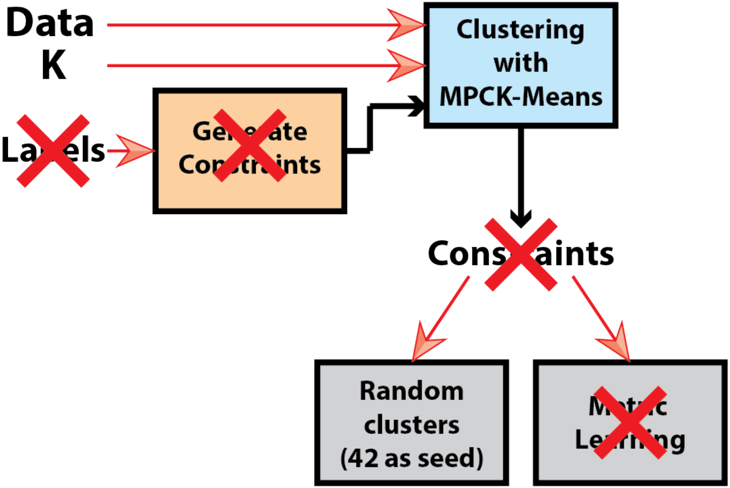
RODA adaptation to other experimental procedures

Clustering



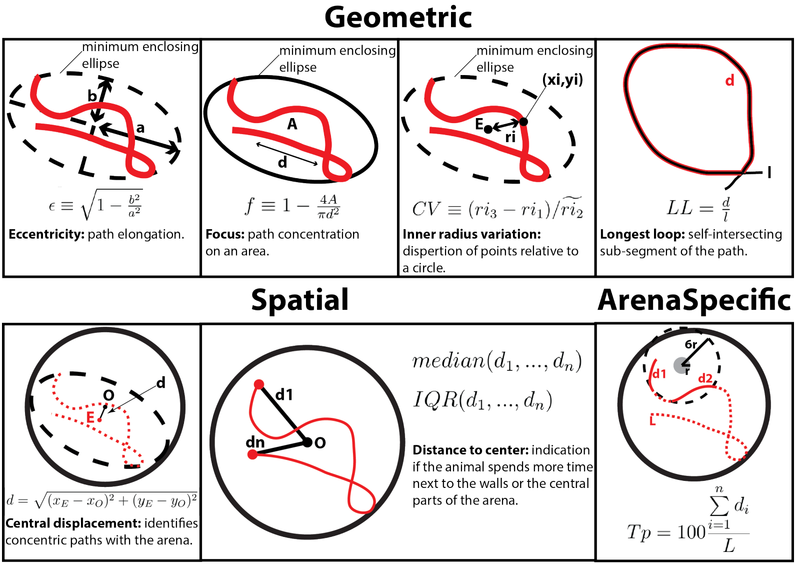
RODA adaptation to other experimental procedures

Clustering



RODA adaptation to other experimental procedures

Path features



RODA adaptation to other experimental procedures

Overlapping segmentation

- Generates huge amount of data.
- Creates difficult to separate data.
- It cannot capture stationary points.

RODA adaptation to other experimental procedures

Solutions

- Implementation of more path features.

RODA adaptation to other experimental procedures

Solutions

- Implementation of more path features.
- A generic segmentation criterion which might be combined with the overlapping segmentation (path sinuosity [1]).

[1] Benhamou, Simon. "How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension?." Journal of theoretical biology 229.2 (2004): 209-220.

RODA adaptation to other experimental procedures

Solutions

- Implementation of more path features.
- A generic segmentation criterion which might be combined with the overlapping segmentation (path sinuosity [1]).
- Clustering:
  - Initialize clusters deterministically based on data density (DKMeans++ [2] →).

[1] Benhamou, Simon. "How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension?." Journal of theoretical biology 229.2 (2004): 209-220.
[2] Nidheesh, N., KA Abdul Nazeer, and P. M. Ameer. "An enhanced deterministic K-Means clustering algorithm for cancer subtype prediction from gene expression data." Computers in biology and medicine 91 (2017): 213-221.

RODA adaptation to other experimental procedures

Solutions

- Implementation of more path features.
- A generic segmentation criterion which might be combined with the overlapping segmentation (path sinuosity [1]).
- Clustering:
  - Initialize clusters deterministically based on data density (DKMeans++ [2] →).
  - Hierarchical clustering (Bisecting K-Means [3]).

[1] Benhamou, Simon. "How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension?." Journal of theoretical biology 229.2 (2004): 209-220.
[2] Nidheesh, N., KA Abdul Nazeer, and P. M. Ameer. "An enhanced deterministic K-Means clustering algorithm for cancer subtype prediction from gene expression data." Computers in biology and medicine 91 (2017): 213-221.
[3] Steinbach, Michael, George Karypis, and Vipin Kumar. "A comparison of document clustering techniques." KDD workshop on text mining. Vol. 400. No. 1. 2000.

RODA adaptation to other experimental procedures

Solutions

- Implementation of more path features.
- A generic segmentation criterion which might be combined with the overlapping segmentation (path sinuosity [1]).
- Clustering:
    - Initialize clusters deterministically based on data density (DKMeans++ [2] →).
    - Hierarchical clustering (Bisecting K-Means [3]).
    - Feature weighting based on outliers detection and exclusion [4].

[1] Benhamou, Simon. "How to reliably estimate the tortuosity of an animal's path:: straightness, sinuosity, or fractal dimension?." Journal of theoretical biology 229.2 (2004): 209-220.
[2] Nidheesh, N., KA Abdul Nazeer, and P. M. Ameer. "An enhanced deterministic K-Means clustering algorithm for cancer subtype prediction from gene expression data." Computers in biology and medicine 91 (2017): 213-221.
[3] Steinbach, Michael, George Karypis, and Vipin Kumar. "A comparison of document clustering techniques." KDD workshop on text mining. Vol. 400. No. 1. 2000.
[4] Brodinova, Sarka, et al. "Robust and sparse k-means clustering for high-dimensional data." arXiv preprint arXiv:1709.10012 (2017).

Thank you for your attention!

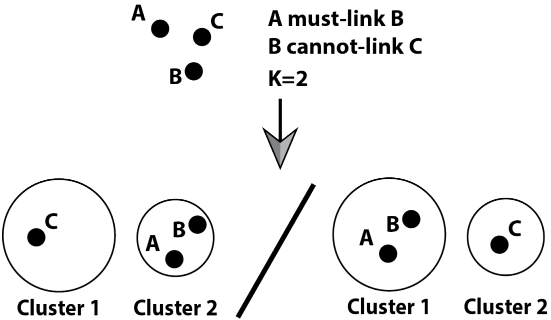
Any questions?

Metric Pairwise-Constrained K-Means (MPCK-Means)

Bilenko, Mikhail, Sugato Basu, and Raymond J. Mooney. "Integrating constraints and metric learning in semi-supervised clustering." Proceedings of the twenty-first international conference on Machine learning. ACM, 2004.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Pairwise constraints



Example: The COP-KMeans; constraints are never broken when updating cluster assignments [1].

[1] Wagstaff, Kiri, et al. "Constrained k-means clustering with background knowledge." ICML. Vol. 1. 2001.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Metric learning

$$d_A(x_1, x_2) = \|x_1 - x_2\|_A = \sqrt{(x_1 - x_2)^T A (x_1 - x_2)} \quad (1)$$

- if $A = I$ then (1) corresponds to the Euclidean distance.
- if A is diagonal matrix and not I then each axis or dimension is given a weight (feature weighting).
- if A is full matrix then new features are generated that are linear combination of the original features [2].

[1] Xing, Eric P., et al. "Distance metric learning with application to clustering with side-information." Advances in neural information processing systems. 2003.
[2] Basu, Sugato, Mikhail Bilenko, and Raymond J. Mooney. "Comparing and unifying search-based and similarity-based approaches to semi-supervised clustering." Proceedings of the ICML-2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining. 2003.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Initialize cluster centroids

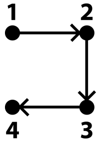
- Create λ neighborhoods by using the transitive closure of the MUST-LINK constraints.

skip explanation

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Initialize cluster centroids

- Create λ neighborhoods by using the transitive closure of the MUST-LINK constraints. skip explanation

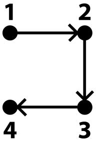


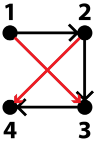
The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Initialize cluster centroids

- Create λ neighborhoods by using the transitive closure of the MUST-LINK constraints.

skip explanation



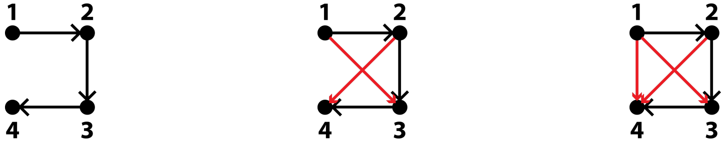


The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Initialize cluster centroids

- Create λ neighborhoods by using the transitive closure of the MUST-LINK constraints.

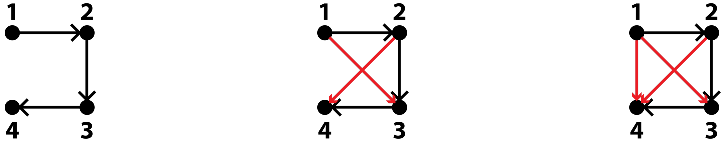
skip explanation



The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Initialize cluster centroids

- Create λ neighborhoods by using the transitive closure of the MUST-LINK constraints. skip explanation



▷ $L = \{(1,2),(2,3),(3,4)\}$
$\oplus \{(1,3),(2,4)\}$
$\oplus \{(1,4)\}$

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

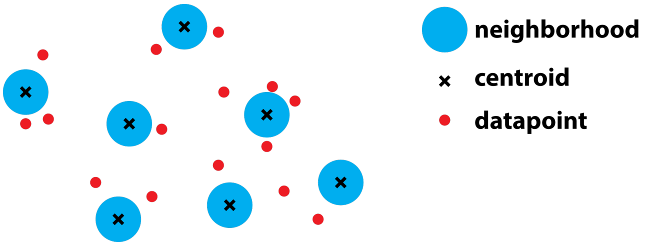
Initialize cluster centroids

- Create λ neighborhoods by using the transitive closure of the MUST-LINK constraints.
- Augment the MUST-LINK and CANNOT-LINK sets of constraints with any additional constraints.
- Use the centers of the neighborhoods as centroids:
  - if $k = \lambda$ initialize λ centroids.
  - if $k > \lambda$ initialize λ centroids and the remaining $k - \lambda$ centroids at random *using* 42 *as random seed*.
  - if $k < \lambda$ initialize k neighborhoods from λ based on weighted farthest-first traversal where the weights are the sizes of the neighborhoods.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

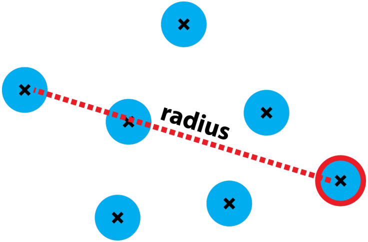
(Weighted) farthest-first traversal

Goal: find K points which are maximally separated from each other (in terms of a weighted distance).



The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

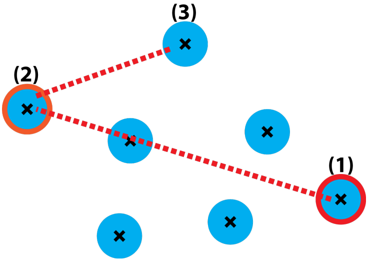
(Weighted) farthest-first traversal



- Pick a neighborhood at random λ_1
- Find the furthest neighborhood of λ_1.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

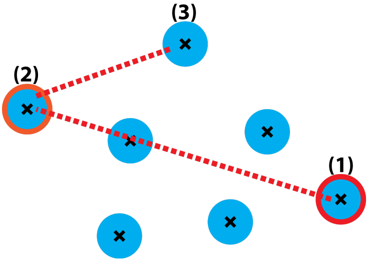
(Weighted) farthest-first traversal



- Find the furthest neighborhood of λ_2 that is also the farthest from the neighborhood λ_1.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

(Weighted) farthest-first traversal



- Find the furthest neighborhood of λ_2 that is also the farthest from the neighborhood λ_1.
- Since *weights* $\equiv$ *size*(λ), the selected points are far apart and inside large neighborhoods.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Integrating constraints and metric learning

$$J = \sum_{x_i \in X} (\|x_i - \mu_{l_i}\|^2_{A_{l_i}} - log(det(A_{l_i}))) \quad (1)$$

$$+ \sum_{(x_i,x_j) \in M} w_{ij} f_M(x_i, x_j) \mathbb{1}[l_i \neq l_j] \quad (2)$$

$$+ \sum_{(x_i,x_j) \in C} \overline{w}_{ij} f_C(x_i, x_j) \mathbb{1}[l_i = l_j] \quad (3)$$

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Integrating constraints and metric learning

$$J = \sum_{x_i \in X} (\|x_i - \mu_{l_i}\|^2_{A_{l_i}} - log(det(A_{l_i}))) \quad (1)$$

$$+ \sum_{(x_i, x_j) \in M} w_{ij} f_M(x_i, x_j) \mathbb{1}[l_i \neq l_j] \quad (2)$$

$$+ \sum_{(x_i, x_j) \in C} \overline{w}_{ij} f_C(x_i, x_j) \mathbb{1}[l_i = l_j] \quad (3)$$

(1) results in the learning of the diagonal matrix A.
(2) is the penalty cost of violating the MUST-LINK constraints.
(3) is the penalty cost of violating the CANNOT-LINK constraints.

The Metric Pairwise-Constrained K-Means (MPCK-Means) algorithm

Integrating constraints and metric learning

$$J = \sum_{x_i \in X} (\|x_i - \mu_{l_i}\|^2_{A_{l_i}} - log(det(A_{l_i}))) \quad (1)$$

$$+ \sum_{(x_i, x_j) \in M} w_{ij} f_M(x_i, x_j) \mathbb{1}[l_i \neq l_j] \quad (2)$$

$$+ \sum_{(x_i, x_j) \in C} \overline{w}_{ij} f_C(x_i, x_j) \mathbb{1}[l_i = l_j] \quad (3)$$

(1) results in the learning of the diagonal matrix A.
(2) is the penalty cost of violating the MUST-LINK constraints.
(3) is the penalty cost of violating the CANNOT-LINK constraints.

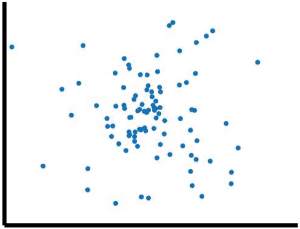
- Severity of M: $f_M = \frac{1}{2}\|x_i - x_j\|^2_{A_{l_i}} + \frac{1}{2}\|x_i - x_j\|^2_{A_{l_j}}$
- Severity of C: $f_M = \|x'_{l_i} - x''_{l_i}\|^2_{A_{l_i}} + \|x_i - x_j\|^2_{A_{l_i}}$, where x'_{l_i} and x''_{l_i} is the maximally separated pair of points in the dataset.

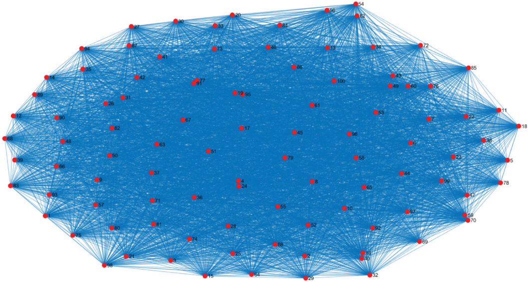
Density K-Means++ (DKM++)

Nidheesh, N., KA Abdul Nazeer, and P. M. Ameer. "An enhanced deterministic K-Means clustering algorithm for cancer subtype prediction from gene expression data." Computers in biology and medicine 91 (2017): 213-221.

The Density K-Means++ (DKM++) algorithm

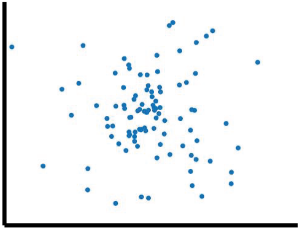
Minimum spanning tree

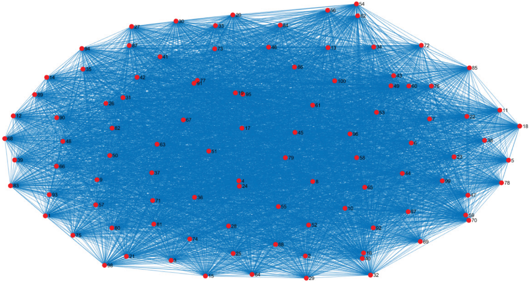




The Density K-Means++ (DKM++) algorithm

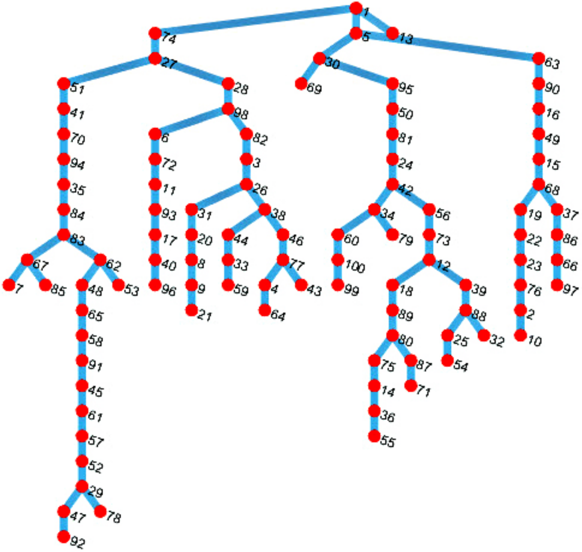
Minimum spanning tree





- Subset of the edges that connects all the vertices together without any cycle and with the minimum weight.

Minimum spanning tree

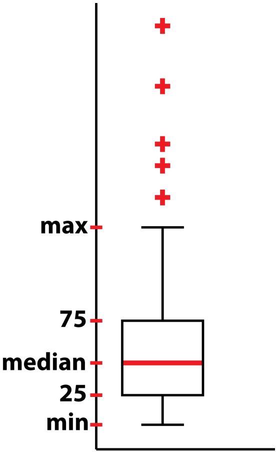


The Density K-Means++ (DKM++) algorithm

Radius using MST-Heuristic

$\epsilon = 3 * IQR(L) + 75^{th} percentile(L)$,

$L \equiv$ MST weights (lengths)

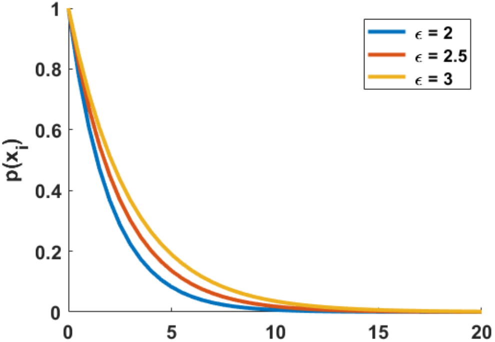


The Density K-Means++ (DKM++) algorithm

Local density

- Find the $\epsilon - neighbors(x_i)$.
- Compute the local density

$$\rho(x_i) = \sum_{y \in \epsilon - neighbors(x_i)} exp(\frac{-\|x_i - y_j\|}{\epsilon})$$

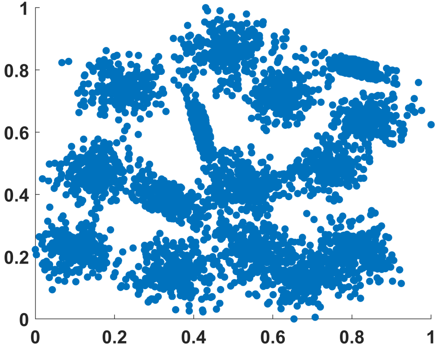


The Density K-Means++ (DKM++) algorithm

Prospectiveness

- $C \longleftarrow \{max(\rho(x))\}$.
- $\phi(x_j) = \rho(x_j) * \|x_j - x_m\|$, x_m is the nearest data point added in C.
- $C \longleftarrow \{max(\rho(x)), max(\phi(x))\}$
- Repeat least 2 steps until k centroids are picked.

The Density K-Means++ (DKM++) algorithm



The Density K-Means++ (DKM++) algorithm