Evolving Controllably Difficult Datasets for Clustering

GECCO 2019

Cameron Shand[1], Richard Allmendinger[2], Julia Handl[2], Andrew Webb[1], & John Keane[1]

[1]*School of Computer Science, University of Manchester*
[2]*Alliance Manchester Business School, University of Manchester*

Generating Problems for Algorithm Selection

Algorithm selection problem

- Predict which algorithm in a portfolio will perform best[1]

[1] Rice, J. R. (1976). The algorithm selection problem. In Advances in computers (Vol. 15, pp. 65-118). Elsevier.

Algorithm selection problem

- Predict which algorithm in a portfolio will perform best[1]
- Requires a range of features that represent the problems/instances[2]

[1] Rice, J. R. (1976). The algorithm selection problem. In Advances in computers (Vol. 15, pp. 65-118). Elsevier.

[2] Smith-Miles, K., & Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. Computers & Operations Research, 39(5), 875-889.

Algorithm selection problem

- Predict which algorithm in a portfolio will perform best[1]
- Requires a range of features that represent the problems/instances[2]
- Learn mapping between problem features and algorithmic performance

[1]Rice, J. R. (1976). The algorithm selection problem. In Advances in computers (Vol. 15, pp. 65-118). Elsevier.

[2]Smith-Miles, K., & Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. Computers & Operations Research, 39(5), 875-889.

Algorithm selection problem

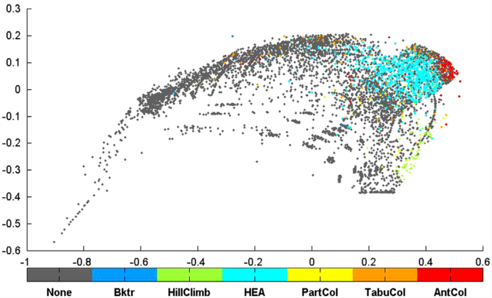
- Predict which algorithm in a portfolio will perform best[1]
- Requires a range of features that represent the problems/instances[2]
- Learn mapping between problem features and algorithmic performance
- Need problems with diverse features to learn mapping

[1] Rice, J. R. (1976). The algorithm selection problem. In Advances in computers (Vol. 15, pp. 65-118). Elsevier.

[2] Smith-Miles, K., & Lopes, L. (2012). Measuring instance difficulty for combinatorial optimization problems. Computers & Operations Research, 39(5), 875-889.

Instance space

Visualization of problem instances to identify algorithmic differential performance



Source: Smith-Miles, K., Baatar, D., Wreford, B., & Lewis, R. (2014). Towards objective measures of algorithm performance across instance space. Computers & Operations Research, 45, 12-24.

Generating instances

- Create datasets with properties not currently exhibited

Generating instances

- Create datasets with properties not currently exhibited
- Datasets across a gradient of difficulty

Generating instances

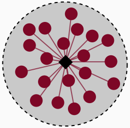
- Create datasets with properties not currently exhibited
- Datasets across a gradient of difficulty
- Requires a flexible generating mechanism

Generating instances

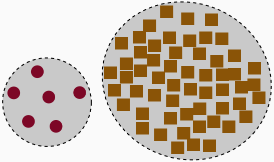
- Create datasets with properties not currently exhibited
- Datasets across a gradient of difficulty
- Requires a flexible generating mechanism
- May require optimization to generate specific properties

Generating Synthetic Clusters

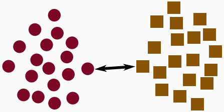
Cluster properties & challenges



Compactness



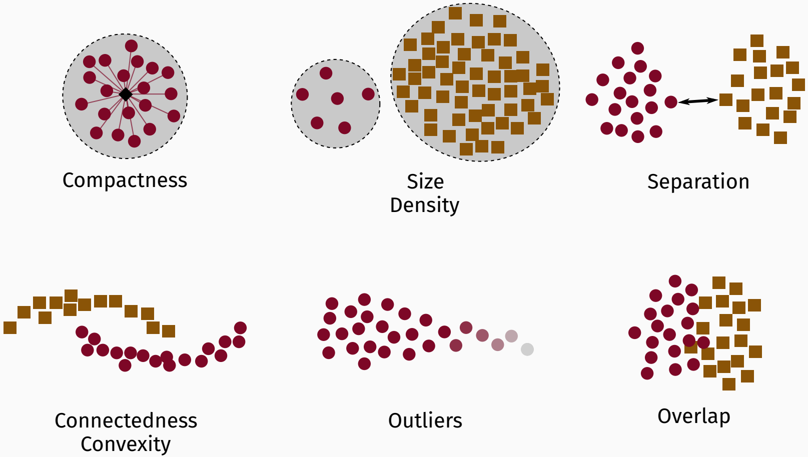
Size
Density



Separation

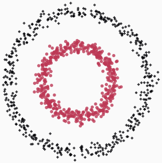
Inspiration from: Handl, J., & Knowles, J. (2006). Multi-objective clustering and cluster validation. In Multi-Objective Machine Learning (pp. 21-47). Springer, Berlin, Heidelberg.

Cluster properties & challenges

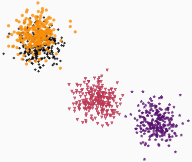


Inspiration from: Handl, J., & Knowles, J. (2006). Multi-objective clustering and cluster validation. In Multi-Objective Machine Learning (pp. 21-47). Springer, Berlin, Heidelberg.

Existing cluster generators



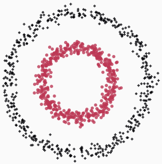
Scikit-learn
'`circles`'[1]



Scikit-learn
'`blobs`'[1]

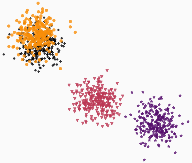
[1] Pedregosa, F., ..., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

Existing cluster generators

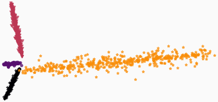


Scikit-learn
'`circles`'[1]

Handl/Knowles (HK)
'*gaussian*'[2]



Scikit-learn
'`blobs`'[1]

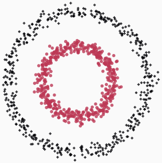


HK
'*ellipsoidal*'[2]

[1] Pedregosa, F., ..., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

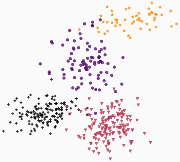
[2] Handl, J., & Knowles, J. (2005). Improvements to the scalability of multiobjective clustering. In 2005 IEEE Congress on Evolutionary Computation (Vol. 3, pp. 2372-2379). IEEE.

Existing cluster generators

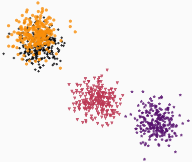


Scikit-learn
'`circles`'[1]

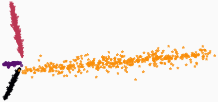
Handl/Knowles (HK)
'*gaussian*'[2]



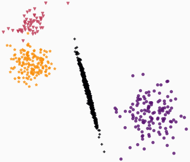
Qiu/Joe (QJ)
generator[3]



Scikit-learn
'`blobs`'[1]



HK
'*ellipsoidal*'[2]



HAWKS
(us)

[1] Pedregosa, F., ..., & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12(Oct), 2825-2830.

[2] Handl, J., & Knowles, J. (2005). Improvements to the scalability of multiobjective clustering. In 2005 IEEE Congress on Evolutionary Computation (Vol. 3, pp. 2372-2379). IEEE.

[3] Qiu, W., & Joe, H. (2006). Generation of random clusters with specified degree of separation. Journal of Classification, 23(2), 315-334.

HAWKS

Handl Allmendinger Webb Keane Shand

~~HAWKS~~

KASH Generator

Fitness of a dataset?

- **Aim:** optimize to a pre-defined value of “difficulty”

[1] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), 243-256.

Fitness of a dataset?

- **Aim:** optimize to a pre-defined value of “difficulty”

- Maximizing e.g. separation would be too simple...

[1] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), 243-256.

Fitness of a dataset?

- **Aim:** optimize to a pre-defined value of “difficulty”
- Maximizing e.g. separation would be too simple…
- Minimizing e.g. separation would be meaningless…

[1] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), 243-256.

Fitness of a dataset?

- **Aim:** optimize to a pre-defined value of “difficulty”
- Maximizing e.g. separation would be too simple...
- Minimizing e.g. separation would be meaningless...
- Using multiple cluster quality measures could get tricky...[1]

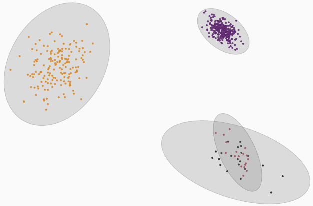
[1] Arbelaitz, O., Gurrutxaga, I., Muguerza, J., Pérez, J. M., & Perona, I. (2013). An extensive comparative study of cluster validity indices. Pattern Recognition, 46(1), 243-256.

Silhouette width issues

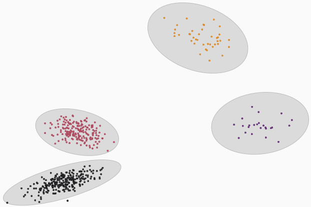
- Silhouette width (*sw*) gives ratio of compactness and separation
- Calculated for each data point, in range $[-1, 1]$

Silhouette width issues

- Silhouette width (sw) gives ratio of compactness and separation
- Calculated for each data point, in range $[-1, 1]$
- Average over data can lead to minima — both examples below have $sw = 0.9$



Example 1



Example 2

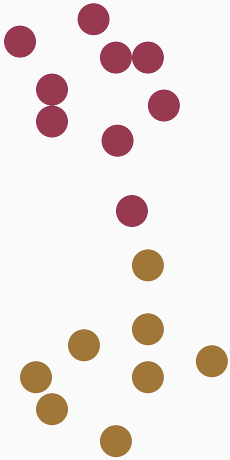
Fitness of a dataset!

- Use pre-defined silhouette width as the target (s_t)
- Gives rough indication of “clusterability”
- Minimize difference between individual’s sw and s_t

$$\min \; f(\mu_1, \Sigma_1, \ldots, \mu_K, \Sigma_K) = |s_t - sw|$$

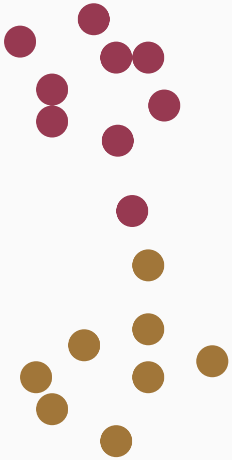
Augmenting difficulty - overlap

Percentage of points whose nearest neighbour is in a different cluster



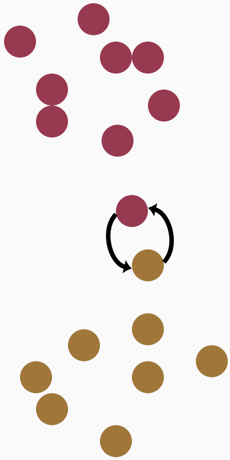
Augmenting difficulty - overlap

Percentage of points whose nearest neighbour is in a different cluster



Augmenting difficulty - overlap

Percentage of points whose nearest neighbour is in a different cluster

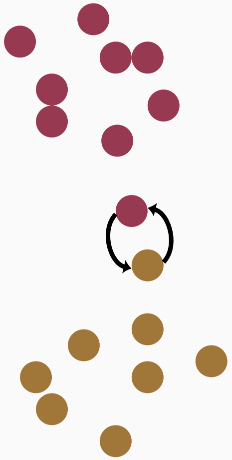


Augmenting difficulty - overlap

Percentage of points whose nearest neighbour is in a different cluster

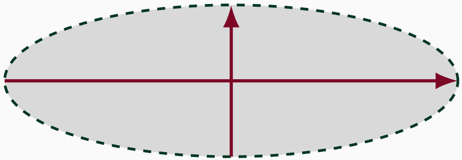
$$overlap = 1 - \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{C^i}(i_{nn})$$

$$\mathbb{1}_{C^i}(\underbrace{i_{nn}}_{i\text{'s nearest neighbour}}) := \begin{cases} 1, & i_{nn} \in C^i \\ 0, & i_{nn} \notin \underbrace{C^i}_{i\text{'s cluster}} \end{cases}$$



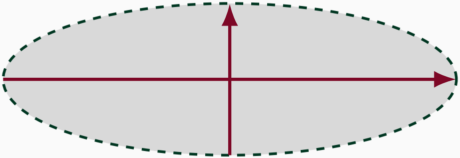
Augmenting difficulty - eccentricity

Ratio of largest to smallest principal axis



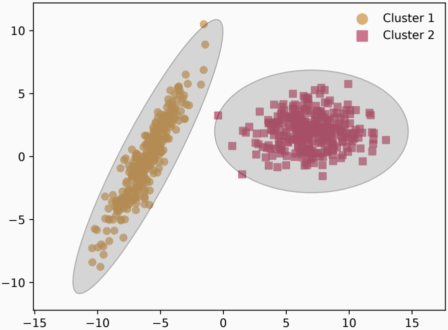
Augmenting difficulty - eccentricity

Ratio of largest to smallest principal axis



$$\lambda^{ratio} = \max_{\forall\, k \in \{1,\ldots,K\}} \frac{\max \forall\, i \in \{1,\ldots,D\} \Sigma_{ii}^{k}}{\min \forall\, i \in \{1,\ldots,D\} \Sigma_{ii}^{k}}$$

Cluster representation



$$\boxed{\begin{bmatrix} -6 & 0 \end{bmatrix}}\boxed{\begin{bmatrix} 3 & 5 \\ 5 & 10 \end{bmatrix}}\boxed{\begin{bmatrix} 7 & 2 \end{bmatrix}}\boxed{\begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}}$$

$$\mu_1 \qquad \Sigma_1 \qquad \mu_2 \qquad \Sigma_2$$

- K Gaussians are specified
- Encoded as the means (μ) and covariances (Σ)
- Each μ represents D variables
- Each Σ represents $D \times D$ variables

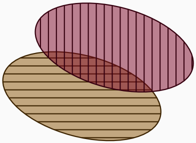
How do you randomly perturb a cluster?

Deal with μ and Σ separately

How do you randomly perturb a cluster?

Deal with μ and Σ separately

Mean

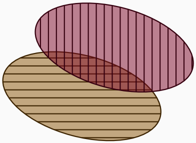


Sample new mean from standard Gaussian around current mean

How do you randomly perturb a cluster?

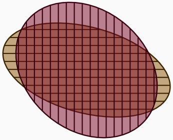
Deal with μ and Σ separately

Mean



Sample new mean from standard Gaussian around current mean

Covariance

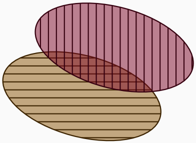


Randomly rotate covariance

How do you randomly perturb a cluster?

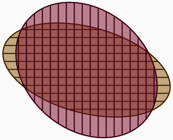
Deal with μ and Σ separately

Mean



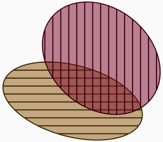
Sample new mean from standard Gaussian around current mean

Covariance



Randomly rotate covariance

Combined



Either, neither, or both can occur

Fitness vs constraints

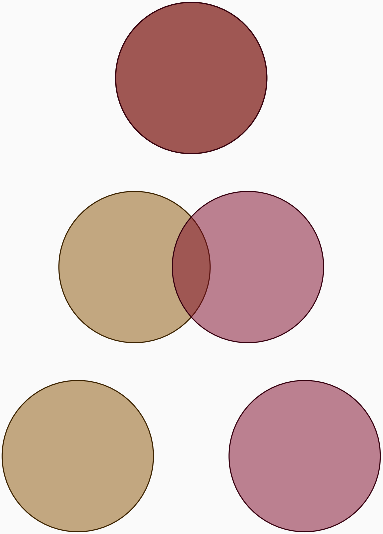
Stochastic ranking (simplified)

```
for i in num_sweeps:
    for j in pop_size:
        I_1 = pop[j]
        I_2 = pop[j+1]
        u = random(0,1)
        if (I_1^feasible and I_2^feasible) or u < P_fitness:
            if I_1^fitness > I_2^fitness:
                swap(I_1,I_2)
        else if I_1^penalty > I_2^penalty:
            swap(I_1,I_2)
    if no swaps:
        break
```

Runarsson, T. P., & Yao, X. (2000). Stochastic ranking for constrained evolutionary optimization. IEEE Transactions on evolutionary computation, 4(3), 284-294.

Experiments

Silhouette width & dimensionality — setup

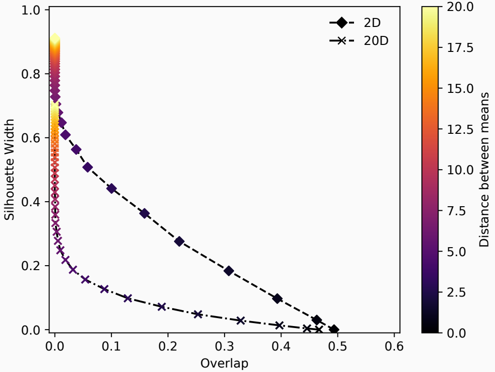


Start with overlapping Gaussians

Gradually separate in 1 dimension

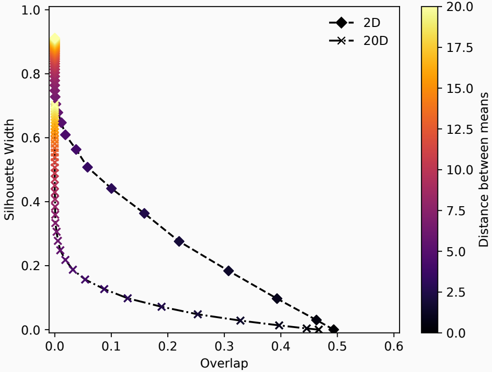
Until well-separated

Silhouette width & dimensionality — results



- Gradual increase in silhouette width alongside overlap decrease in 2D

Silhouette width & dimensionality — results



- Gradual increase in silhouette width alongside overlap decrease in 2D
- Little initial change in silhouette width with overlap decrease in 20D

Balancing fitness and constraints — setup

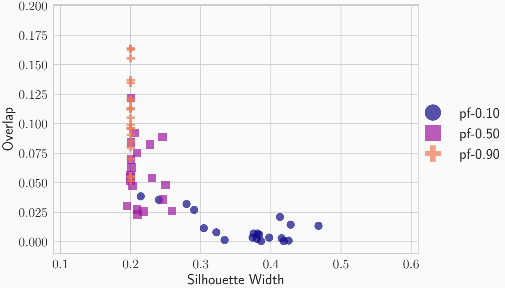
- **Aim**: Vary $P_{fitness}$ with conflicting silhouette width and overlap
- How is the search affected?
- Does it result in different algorithmic performance?

Balancing fitness and constraints — setup

- **Aim**: Vary $P_{fitness}$ with conflicting silhouette width and overlap
- How is the search affected?
- Does it result in different algorithmic performance?

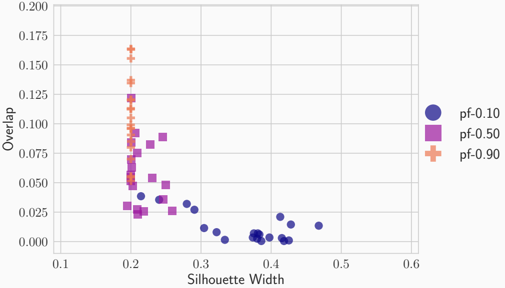
Parameter	Value(s)
$P_{fitness}$	$\{0.1, 0.5, 0.9\}$
s_t	0.2 (20D) 0.6 (2D)
Overlap	≤ 0
Eccentricity	Unconstrained

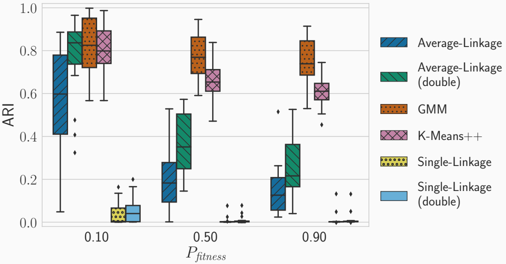
Balancing fitness and constraints — results



- $P_{fitness}$ did clearly affect which solutions were favoured (20D shown)

Balancing fitness and constraints — results





- $P_{fitness}$ did clearly affect which solutions were favoured (20D shown)
- Performance was best with low overlap
- GMM was most robust to increasing overlap

Generator comparison — setup

- **Aim**: Generate a set of diverse datasets
- Does changing s_t actually affect algorithmic performance?
- How does the difference compare to other generators?

Generator comparison — setup

- **Aim**: Generate a set of diverse datasets
- Does changing s_t actually affect algorithmic performance?
- How does the difference compare to other generators?

Parameter	Value(s)
K	$\{5, 30\}$
D	$\{2, 20\}$
$P_{fitness}$	0.5
s_t	$\{0.2, 0.5, 0.8\}$
Overlap	≤ 0
Eccentricity	Unconstrained

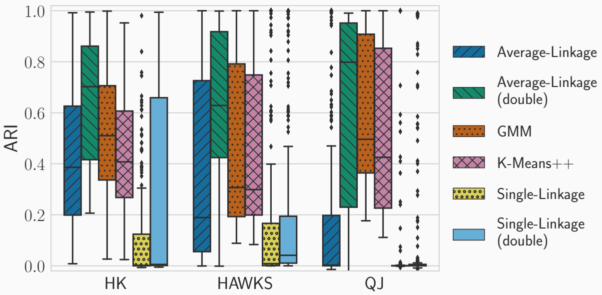
Generator comparison — setup

- **Aim**: Generate a set of diverse datasets
- Does changing s_t actually affect algorithmic performance?
- How does the difference compare to other generators?

Parameter	Value(s)
K	$\{5, 30\}$
D	$\{2, 20\}$
$P_{fitness}$	0.5
s_t	$\{0.2, 0.5, 0.8\}$
Overlap	≤ 0
Eccentricity	Unconstrained

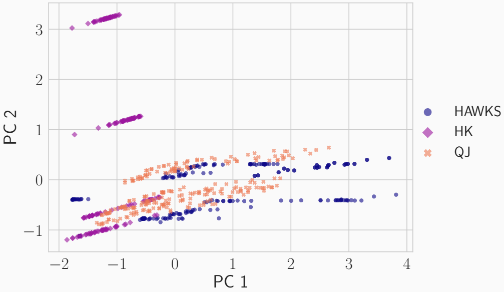
Generator	# Datasets
HAWKS	240
QJ	243
HK	160

Generator comparison — results



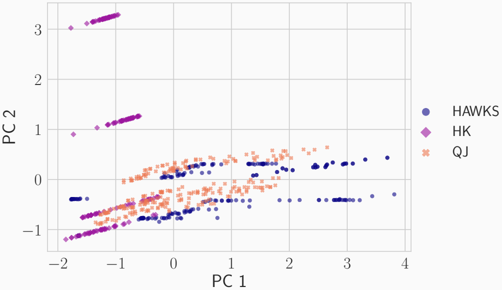
- Spread of results across the generators
- QJ more favoured to compactness-based algorithms
- HK somewhat better at linkage-based
- HAWKS has reasonable performance range

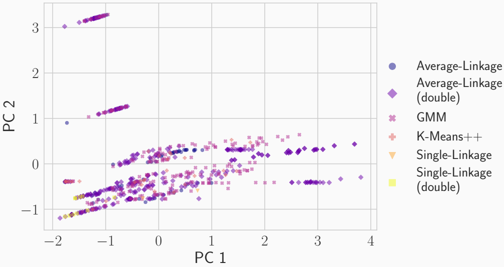
Generator comparison — instance space?



- Simple instance space from 3 features
- Similarly covered space from QJ & HAWKS

Generator comparison — instance space?





- Simple instance space from 3 features
- Similarly covered space from QJ & HAWKS
- Some areas show preference
- Mostly dominated by GMM/Average-linkage

Summary & Future Work

Summary

- Synthetic data is needed in clustering for insightful empirical comparison

Summary

- Synthetic data is needed in clustering for insightful empirical comparison
- Our tool, HAWKS, allows for the generation of datasets of various complexity

Summary

- Synthetic data is needed in clustering for insightful empirical comparison
- Our tool, HAWKS, allows for the generation of datasets of various complexity
- Further work is needed to expand the complexities that can be introduced to the data

Future work

- Non-convex clusters

Future work

- Non-convex clusters
- Multiple objectives

Future work

- Non-convex clusters
- Multiple objectives
- Inclusion other cluster properties/challenges

Future work

- Non-convex clusters
- Multiple objectives
- Inclusion other cluster properties/challenges
- Benchmark suite construction

Questions?

Code available via:

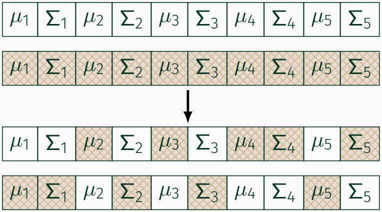
github.com/sea-shunned/hawks

or

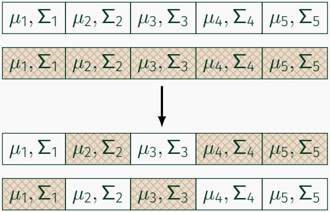
`pip install hawks`

Crossover

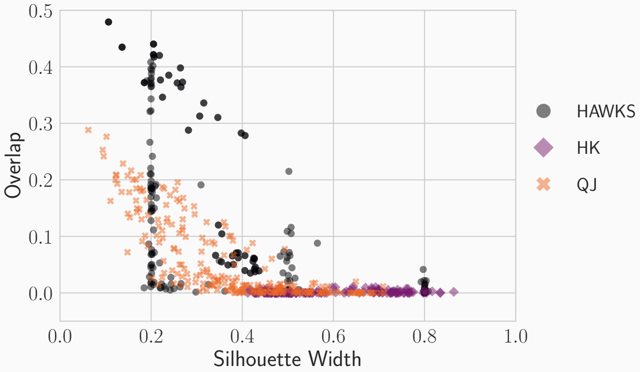
Freely exchange μ and Σ separately



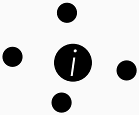
Exchange μ and Σ together

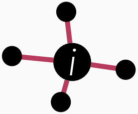


Generator Datasets Silhouette & Overlap



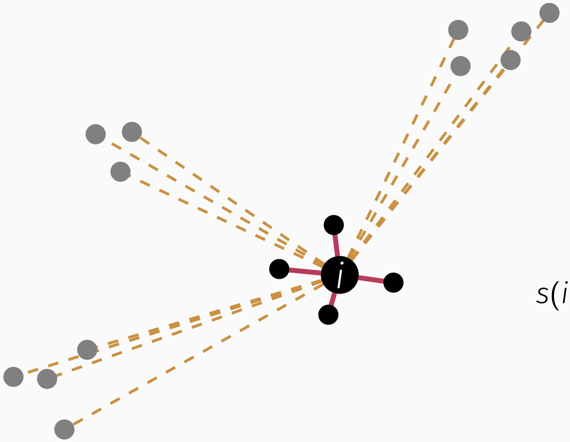
Silhouette width





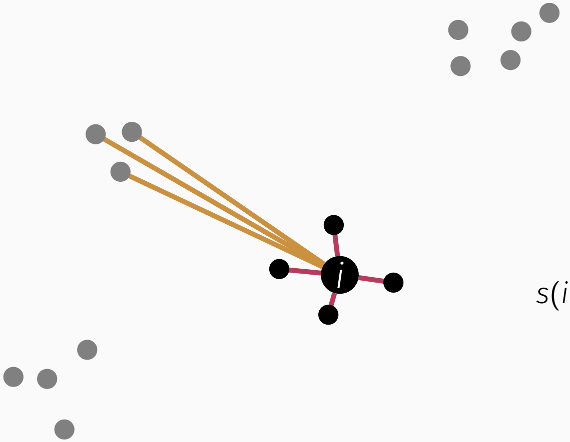
$$s(i) = \frac{\qquad - a(i)}{max\{a(i),\qquad}$$

Silhouette width



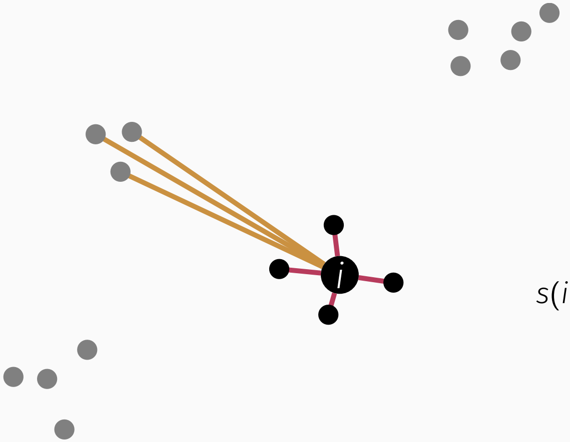
$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Silhouette width



$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

Silhouette width



$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}$$

$$s_{all} = \frac{1}{N}\sum_{i=1}^{N} s(i)$$

“Real” vs synthetic data

- Real-world data allows us to directly test algorithmic applicability[1]

[1] Von Luxburg, U., Williamson, R. C., & Guyon, I. (2012, June). Clustering: Science or art?. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 65-79).

[2] Hooker, J. N. (1995). Testing heuristics: We have it all wrong. Journal of heuristics, 1(1), 33-42.

[3] Macià, N., & Bernadó-Mansilla, E. (2014). Towards UCI+: a mindful repository design. Information Sciences, 261, 237-262.

"Real" vs synthetic data

- Real-world data allows us to directly test algorithmic applicability[1]

- Difficult to make sufficiently complex synthetic data

[1] Von Luxburg, U., Williamson, R. C., & Guyon, I. (2012, June). Clustering: Science or art?. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 65-79).

[2] Hooker, J. N. (1995). Testing heuristics: We have it all wrong. Journal of heuristics, 1(1), 33-42.

[3] Macià, N., & Bernadó-Mansilla, E. (2014). Towards UCI+: a mindful repository design. Information Sciences, 261, 237-262.

"Real" vs synthetic data

- Real-world data allows us to directly test algorithmic applicability[1]

- Difficult to make sufficiently complex synthetic data

- Synthetic data has corresponding generating model

[1] Von Luxburg, U., Williamson, R. C., & Guyon, I. (2012, June). Clustering: Science or art?. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 65-79).

[2] Hooker, J. N. (1995). Testing heuristics: We have it all wrong. Journal of heuristics, 1(1), 33-42.

[3] Macià, N., & Bernadó-Mansilla, E. (2014). Towards UCI+: a mindful repository design. Information Sciences, 261, 237-262.

“Real” vs synthetic data

- Real-world data allows us to directly test algorithmic applicability[1]
- Difficult to make sufficiently complex synthetic data
- Synthetic data has corresponding generating model
- Benchmarks allow for community-wide comparisons, with a range of data properties[2,3]

[1] Von Luxburg, U., Williamson, R. C., & Guyon, I. (2012, June). Clustering: Science or art?. In Proceedings of ICML Workshop on Unsupervised and Transfer Learning (pp. 65-79).

[2] Hooker, J. N. (1995). Testing heuristics: We have it all wrong. Journal of heuristics, 1(1), 33-42.

[3] Macià, N., & Bernadó-Mansilla, E. (2014). Towards UCI+: a mindful repository design. Information Sciences, 261, 237-262.