Effects of hyper-parameters in online constrained clustering: A study on animal videos

F.J. WILLIAMS and L.I. KUNCHEVA, Bangor University, United Kingdom

The aim of online clustering is to discover a structure in running data. Adding label constraints or pairwise constraints to this has shown to improve the clustering accuracy. In this study we present an analysis of how different hyperparameters – proportion of constraints, initial number of clusters, and batch window size – affect most recent and popular online constrained clustering methods, using three different metrics. Our results show that initial number of clusters and window size have an effect on clustering results, while the proportion of constraints does not. We also demonstrate that online clustering performs better than clustering of the whole data together. Our overall findings point at the need for new, more effective online constrained clustering methods.

CCS Concepts: • **Computing methodologies → Object identification**; **Tracking**; **Scene understanding**; **Image representations**.

Additional Key Words and Phrases: constrained clustering, incremental clustering, online clustering, sequential clustering

ACM Reference Format:
F.J. Williams and L.I. Kuncheva. 2023. Effects of hyper-parameters in online constrained clustering: A study on animal videos. In *Proceedings of The 4th Symposium on Pattern Recognition and Applications (SPRA 2023).* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Object re-identification in videos is an ongoing challenge. In order to reduce the need for manual processing of video data, automated methods have been formulated, based on clustering and classification. Most of the work thus far has been focused on people face detection and vehicle detection [14, 23, 24, 34]. Object Classification relies on a pre-trained model which determines the identity of an object based on what it has been trained to detect. Object Clustering is used where training of a bespoke model is impractical or unfeasible, by instead partitioning the data into clusters which are collated based on some pre-defined features.

We will be focusing on clustering, specifically *Online Constrained Clustering*. This method has two additions to 'plain' clustering - processing the data in batches, and integrating constraint-based knowledge about the data to supplement the partitioning process. This in turn adds two advantages to the original method - the ability to cluster the data in real time, and having the option of aiding the algorithm to achieve a better clustering of the data. While online constrained clustering has been applied (albeit without an explicit formulation) in re-identification faces in well-structured videos such as TV clips [3, 5, 16], to the best of our knowledge, there are no studies on online constrained clustering of unrestricted animal videos. In this study we offer such an experiment using a publicly available video database. The videos are not restricted, which means that there are no defined

SPRA 2023, December 01–03, 2023, Napoli, Italy

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

'shots', and no requirement that the camera is static. Since there are no mainstream models for online constrained clustering, we adapt the existing method for human face re-identification, and add versions of online COP-Kmeans and online constrained single linkage. Our main aim is to find out how these methods compare to one another and to three trivial (baseline) methods on the animal video. To this end, we carry out an experimental study by varying three parameters: the initial number of clusters, the moving window size, and the number of pairwise constraints.

The rest of this study is organised as follows. Section 2 contains a detailed review of current literature on constrained clustering, online constrained clustering and their applications. In Section 3, we discuss the methodology of the study, which includes a brief description of the datasets used (3.1), the methods used (3.2), the metrics used to evaluate our experiment (3.3), as well as the experimental setup (3.4). Finally, the results are presented in Section 4.

2 RELATED WORK

Constrained Clustering (CC) is a field which has grown considerably in recent years [11]. *Online Constrained Clustering* (Online CC), is the focus of this study. In this method, also known as *Sequential* or *Incremental* CC, data comes in as a stream or batch and is clustered as such, rather than as a whole like in standard CC.

Online CC has been used in video [5, 10, 16, 21], still images [9, 35], and on synthetic data [2, 12, 26]. For comparison, we also discuss some standard, offline CC algorithms [3, 25, 28], as well as some unconstrained online clustering methods.

CC has an edge over regular unconstrained clustering as it supports the partitioning by providing the algorithm with some knowledge that we may already have on the data. This knowledge usually comes in the form of pairwise constraints, i.e. must-link (ML) and cannot-link (CL) constraints [7, 20, 31]. The first shows pairs of objects which belong in the same cluster, e.g., those in an adjacent video frames whose bounding boxes overlap [16]. The latter represents pairs of objects that cannot be in the same partition, e.g., two objects in the same video frame for whom it is impossible to have the same identity. This example is known as a temporal constraint and can be used for evolving data [25]. Arbitrarily shaped classes in high-dimensional spaces which cannot be clustered well using traditional methods, will also benefit from any prior knowledge [3]. CC has been used, amongst other fields, in face tracking [28], political analysis [25], and vehicle routing [1].

Online clustering is also effective over regular clustering as it clusters objects in their most recent state. For example, as objects move across a video, their general appearance may change significantly after a number of frames. There is thus an argument that the object's former appearance will not help achieve a higher accuracy when the clustering is performed on the overall dataset, and therefore could lead to the same object at different time intervals being assigned to different clusters [4]. Another argument for online clustering is that one can expect a higher certainty of assigning a smaller number of points to an existing cluster [18]. Certain benefits can be associated with online clustering such as increased speed; scalability to larger datasets as only parts of the data are used at one time; discovery of arbitrarily shaped clusters; and better handling of outliers. Applications include face detection [22], text analysis on social media [8], network flows, and analysis of sensor data [6].

Online constrained clustering has been applied to text and object identification [9, 25, 35], and face tracking [5, 10, 16, 21]. Although Rizoiu et al. [25] do not propose a strictly online approach, the constraint selection is performed in an online way. Some online CC methods adapt existing CC methods to take in a data as a batch or datastream [2, 25], while others developed new methods [16].

In view of the lack of studies on online CC for animal re-identification from video, here we carry out an experiment on a database of five unrestricted animal videos. We adapt existing CC

(a) Koi fish (b) Pigeons (Ground) (c) Pigeons (Kerb)

(d) Pigeons (Square) (e) Pigs (f) Example bounding box

Fig. 1. The video datasets used in the study (a) – (e), and an example of a bounding box, (f).

methods to fit the online framework and use directly an existing Online CC method for human face re-identification.

3 METHODOLOGY

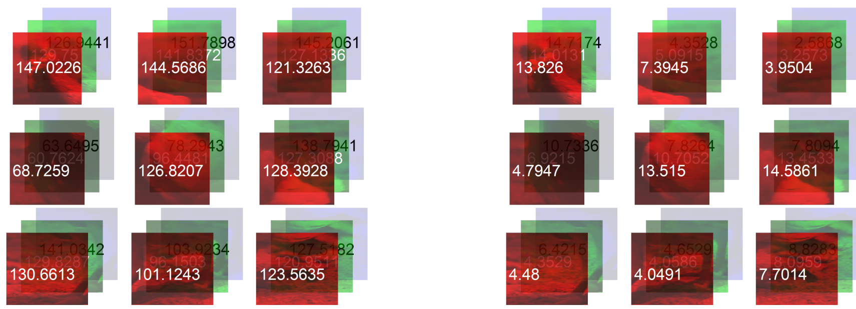
This section describes the data used, the online CC methods, the evaluation metrics, as well as how the experiment is carried out.

3.1 Data

The dataset used in this study consists of feature sets from five videos of animals including koi fish (9 identities); pigeons on the ground (17 identities), a kerb (16 identities), and a square (28 identities); and pigs (26 identities). Fig. 1 shows sample frames from the dataset [1], as well as an example of a bounding box containing one animal. The videos' lengths range from 300-600 frames (9-24 seconds) with frame rates between 24-30 fps. From each video, each animal in every frame was extracted to a unique bounding box. Initially, we extracted different types of features from each bounding box: RGB moments; the code layer of 10 nodes from an autoencoder neural network, the histograms of oriented gradients (HOG), the local binary patterns (LBP), and the 4096 outputs of the penultimate layer of an MobileNetV2 neural network. The features used in this study are the RGB moments, as these proved to be the best set in a previous study [17]. The features are saved in *.csv* format as an $N \times 54$ matrix, i.e. every Nth object contains the mean (μ) and standard deviation (σ) for each RGB colour value, where each bounding box has been split into a 3-by-3 block. Fig. 2 shows the RGB means and standard deviations of the 9 parts of the bounding box displayed in Fig. 1f.

Ground truth labels are available for each video. These were obtained by manual annotation. Bounding boxes were outlined and labelled in each frame. Subsequently, features were extracted from the bounding boxes.

[1] Dataset available at https://zenodo.org/record/7322821



(a) μ for the three colour spaces

(b) σ for the three colour spaces

Fig. 2. An example of the 54 extracted RGB colour values from the bounding box in Fig. 1f.

3.2 Clustering Methods

We have used popular and recent methods, as well as three 'baseline' methods for comparison. Since the online constrained COP-Kmeans and the online constrained Single Linkage had to be adapted here, we explain them in more detail.

3.2.1 KG Algorithm. This method is an online constrained face clustering algorithm which clusters tracks of faces in videos, in this paper named *KG* after its authors (Kulshreshtha and Guha) [16]. This is done by first separating the video into 'shots' of consecutive frames. Within each shot, tracks of faces are created by grouping detections with an overlap of a predefined threshold in adjacent frames. The tracks are then clustered through the use of a similarity matrix which compares and assigns each track to a suitable cluster centre or, if necessary, creates a new cluster.

3.2.2 Online COP-Kmeans (OCC-Kmeans). Constrained K-means is one of the first implementations of a clustering algorithm incorporating background knowledge [29, 31]. COP-Kmeans works like standard K-means with the exception of the pairwise constraints having to be satisfied before the point is assigned to the nearest cluster. If no such cluster exists, the algorithm stops.

In the online version that we created here, we apply the method to each window. The initial centres in the very first window are the first k points in the window. Denote the current cluster centres by CC. We are using CC returned from the previous window as initial centres for the current window. Denote the new centres obtained by standard COP-Kmeans for the current window by CC'. The current centres to be returned are updated as $CC \leftarrow (CC + CC')/2$. The algorithm returns the labels assigned progressively throughout the online procedure.

3.2.3 Online Single Linkage (OCC-SL). Several methods for constrained hierarchical clustering have been proposed in recent years [19, 32, 33]. However, as we have not been able to identify a suitable *online* constrained clustering single linkage method, we offer an intuitive adaptation of our own. In the first window, we create an initial partition by finding the transitive closure of the ML constraints and subsequently applying COP-Kmeans using the pre-specified number of clusters k. The transitive closure is calculated by creating a graph of all objects with edges corresponding to the ML constraints, and then finding the connected components of the graph. We can think of these connected components as 'tracks' in the window composed of several frames of the video. To apply the OCC-SL method, we used the following steps:

(1) Find the tracks in the first window and apply COP-Kmeans using the pre-specified number of clusters k. Denote the current partition by W.
(2) *For each subsequent window*
  (a) Find the tracks in the window. These will be the clusters that need to be assigned using W as the reference structure. Each single object that does not belong to a track, is a cluster of its own. Denote the partition of the current window by W'.
  (b) For each cluster in W', find the closest cluster in W. If the assignment does not violate any CL constraints, label all points in that cluster accordingly and add them to the reference set W. If there is CL constraint violation, apply the same check to the next closest cluster. Continue until a cluster from W accepts the current cluster from W', or there are no remaining clusters in W' to check. If an accepting cluster cannot be identified, create a new cluster and add it to W.
  (c) Retain only the points from the current window as the new W.
(3) Return the labels assigned progressively throughout the online procedure.

3.2.4 Baseline Methods. In addition to the above methods, we used three baseline methods to verify that online constrained clustering works better than trivial labelling and chance. These are called *Baseline "same"*, *"different"* and *"random"*. These methods return the cluster labels as follows:

- *Same* ∘ All objects are labelled as one cluster. ∘ Returns an $N \times 1$ matrix of the same value, e.g. '1'
- *Different* ∘ Every object is a cluster of its own ∘ Returns an $N \times 1$ matrix containing the values $1 \dots N$
- *Random* ∘ For each sliding window, every object is assigned a random cluster label. ∘ Returns an $N \times 1$ matrix of random integers within $\{1, \dots, ws\}$, where ws is the window size.

3.3 Metrics

In order to gauge the quality of the six methods, the following metrics have been utilised.

3.3.1 Normalised Mutual Information. NMI calculates the common information between two sets of labels, e.g. the assigned labels (U) and true labels (V) [27, 30]. The *entropies* of partitions U and V are calculated as:

$$H(X) = -\sum_{x \in X} p(x) \log p(x) \tag{1}$$

where $p(x)$ is the probability that object x from the set of N data points falls into cluster X, defined as:

$$p(x) = \frac{|X|}{N} \tag{2}$$

From this, the mutual information can be calculated using:

$$I(U, V) = \sum_{u \in U} \sum_{v \in V} p(u, v) \log \frac{p(u, v)}{p(u) \times p(v)} \tag{3}$$

This score is normalised as there is no upper bound to the mutual information calculation:

$$NMI(U, V) = \frac{I(U, V)}{\sqrt{H(U) \times H(V)}} \tag{4}$$

A higher score is more desirable, i.e. 0 shows no similarity between the clusters, while 1 shows a perfect match.

3.3.2 Adjusted Rand Index. The Rand Index is calculated by tallying up all of $\binom{N}{2}$ pairs of object labels in the assigned labels and in the true labels by putting them into four categories [13, 30]:

N_{11}	Pair of objects is in **the same** cluster in **both** U and V
N_{01}	Pair of objects is in **the same** cluster in U but in **different** clusters in V
N_{10}	Pair of objects is in **different** clusters in U but in **the same** cluster in V
N_{00}	Pair of objects is in **different** clusters in **both** U and V

The Rand Index is calculated using:

$$RI(U,V) = (N_{00} + N_{11}) / \binom{N}{2} \tag{5}$$

To correct for chance, the Adjusted Rand index is calculated as follows:

$$ARI(U,V) = \frac{RI(U,V) - E_{RI}(U,V)}{1 - E_{RI}(U,V)}, \tag{6}$$

where $E_{RI}(U,V)$ is the expected value of the Rand index:

$$E_{RI}(U,V) = \frac{\sum_{i(U)} \binom{n_i}{2} \times \sum_{j(V)} \binom{n_j}{2}}{\binom{N}{2}}, \tag{7}$$

where n_p is the number of objects in cluster p in the respective partition.

3.3.3 Classification Accuracy. There are two versions of classification accuracy that have been considered here. We refer to them as 'Count' and 'Hungarian'. Denote again the assigned labels by U and true labels by V.

In the 'Count' version, we inspect the *true labels* in each cluster in U, and match this cluster to the cluster in V where the majority of the objects belong. With this method, many clusters in U can be matched to the same cluster from V.

In the 'Hungarian' version, an optimal assignment is calculated, where one-to-one matching is enforced. We used the Munkres algorithm [15], which allows for different number of clusters in U and V. The unassigned labels in U are counted as errors.

While NMI and ARI are symmetrical with respect to the two partitions, neither of the two versions of the Classification Accuracy is. Between the two, we chose the 'Hungarian' version because the 'Count' version strongly favours large number of clusters in U, thereby giving inadequate results when the number of clusters differs significantly between U and V.

3.4 Experiment Setup

The purpose of the experiment is to determine: (1) Which of the compared methods for online CC fares the best on animal re-identification for our data collection; (2) What combination of hyper-parameters works best for each video and across videos. The hyper-parameters we examined here are:

- k – initial number of clusters
- ws – the window size (number of video frames)
- γ – proportion of constraints relative to the number of objects in the window.

The moving window 'skips' ws frames, so consecutive windows are non-overlapping. By using a fixed number of video frames in a moving window, we allow for different number of objects in each window. This depends on how many animals are present simultaneously in each frame.

The ML constraints are sampled from a total pool of ML constraints obtained in the following way: Find the IoU overlap between each pair of bounding boxed in consecutive frames. All pairs with IoU larger than 0.5 are considered to be ML pairs. The set of CL constraints contains all pairs of objects which are in the same frame.

In order to find the optimum values of k, ws and γ, we have to iterate through all the parameters and run the methods with each combination. The sets of hyper-parameter values were:

(1) k = [5, 8, 11, 13, 16, 19, 22, 24, 27, 30]
(2) ws = [3, 6, 9, 12, 15, 18, 21, 24, 27, 30]
(3) γ = [0.0, 0.1, 0.3, 0.5, 0.7, 0.9]

These values have been chosen because: (1) they contain interval values encompassing the minimum and maximum number of identities (9 - 28); (2) each video has a relatively small number of frames, thus a range of smaller window sizes is used; (3) we included in the γ set a proportion of 0, which will give us another baseline – the plain online CC without any constraints.

To calculate the three metrics, we used the ground truth labels for each video.

MATLAB code is available on GitHub [2].

4 RESULTS

4.1 Best combination of parameters

Table 1 show the highest NMI, ARI, and HCA for each video. The columns contain the optimal k_{ini}, the resulting (k_{last}) number of clusters, the optimal window size (ws), the optimal proportion of constraints (γ), the corresponding metric score, the optimal method. For further information we include the average number of clusters per frame (k_{ave}/frame) and window (k_{ave}/window) for the respective ws, both calculated from the ground truth labels. We can see that a high value of k is preferred in the data, and a low ws is optimal across the board; however, a definite observation cannot be made for γ.

An interesting observation in the table is that the resulting number of clusters for the Pigeons (Kerb) and Pigs videos is similar to the true value for all the metrics. In comparison, the same values are far off for other videos. This shows that the methods used do not split and merge the clusters well on all datasets.

4.2 Comparison between the online constrained clustering methods

Fig. 3 shows the results for the three chosen metrics against the plane spanned by k and ws, for the five videos. The three clustering methods and the three baselines are shown as surfaces of different colours. The surfaces are averaged across the values of the constraint proportion γ. This was deemed acceptable because of the very small variation of the values for different γ. The figure leads to the following observations:

- The surfaces are not flat, which suggests that the combination of parameters k and ws has an effect on the quality of the clustering results, as measured by all three metrics.
- There is no single clustering method which dominates the others for all videos. Interestingly, for different combinations of (k, ws), different clustering methods may be more successful, as seen for video Pigeons (Ground) in the figure. OCC-SL was the worst of the three online constrained clustering methods. This is backed up in Table 1, which shows that the KG method works best for the Koi video, while OCC-Kmeans works best for the remainder of the videos.

[2]MATLAB code available at https://github.com/frankmnb/Hyperparameter-Analysis-in-Online-Constrained-Clustering

Table 1. Best combinations of hyperparameters for all videos and methods. For comparison, the **bold** number in brackets in column k_{last} shows the expected number of clusters.

NMI

Video	k_{ini}	k_{last}	ws	γ	Score	Method	k_{ave} /frame	k_{ave} /window
Koi	11	11 **(9)**	3	0.1	0.69	KG	3.05	3.07
Pigeons (Ground)	27	24 **(17)**	6	0	0.44	KG	5.13	5.17
Pigeons (Kerb)	27	12 **(16)**	3	0	0.58	OCC-Kmeans	10.61	10.85
Pigeons (Square)	30	12 **(28)**	3	0	0.65	OCC-Kmeans	16.31	16.42
Pigs	30	24 **(26)**	6	0	0.55	OCC-Kmeans	12.37	12.39

ARI

Video	k_{ini}	k_{last}	ws	γ	Score	Method	k_{ave} /frame	k_{ave} /window
Koi	11	11 **(9)**	3	0.1	0.56	KG	3.05	3.07
Pigeons (Ground)	8	8 **(17)**	3	0.5	0.23	OCC-Kmeans	5.13	5.17
Pigeons (Kerb)	30	12 **(16)**	6	0	0.40	OCC-Kmeans	10.61	10.85
Pigeons (Square)	30	12 **(28)**	3	0	0.42	OCC-Kmeans	16.31	16.42
Pigs	30	24 **(26)**	6	0	0.39	OCC-Kmeans	12.37	12.39

HCA

Video	k_{ini}	k_{last}	ws	γ	Score	Method	k_{ave} /frame	k_{ave} /window
Koi	22	22 **(9)**	6	0.7	0.69	KG	3.05	3.11
Pigeons (Ground)	8	8 **(17)**	6	0	0.40	OCC-Kmeans	5.13	5.17
Pigeons (Kerb)	27	12 **(16)**	3	0	0.51	OCC-Kmeans	10.61	10.85
Pigeons (Square)	30	12 **(28)**	3	0	0.54	OCC-Kmeans	16.31	16.42
Pigs	30	24 **(26)**	6	0	0.50	OCC-Kmeans	12.37	12.39

- As expected, the three three methods are substantially better than the three baseline methods for all datasets.

For comparison, Table 2 shows the optimum metrics for the offline versions of K-Means and Single Linkage versus the optimum metrics for the online algorithms with $\gamma = 0$. Due to their offline and unconstrained nature the only tunable parameter is k. The same values of k have been used as in the remainder of the experiment. Both methods have been calculated using the built in functions in MATLAB, and evaluated using the three metrics. The table shows that, in the best case, the online algorithms outperform the offline algorithms for all videos. Generally, we would expect a better clustering result from the whole data; however, here we can see that the online component has a positive impact on the results.

It can also be observed from Fig. 3, that the three clustering methods have different surface patterns in the (k, ws) space. The pattern, however, are similar across the different metrics and the different videos. To explore this issue further, we pooled the results for each clustering method using only one of the metrics (NMI). Fig. 4 shows the three methods separately, where each plot contains the NMI surfaces for the five videos.

Since the raw values of any metric are non commensurable between datasets, we calculated instead the ranks of the values in the 10-by-10 matrix (k, ws). The 100 raw NMI values for a given video and method were ranked and then replaced by their ranks. Thus, the best combination of

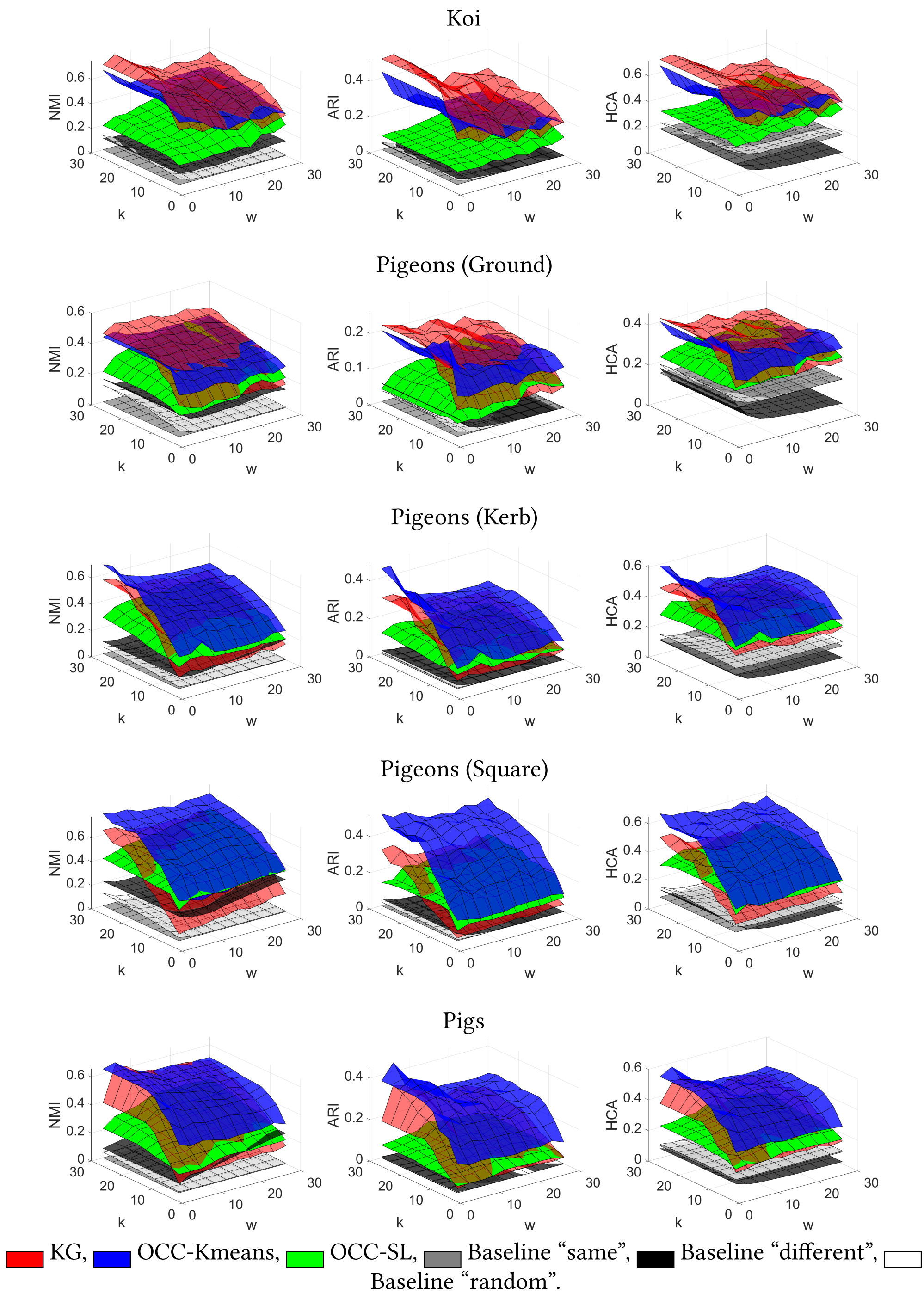
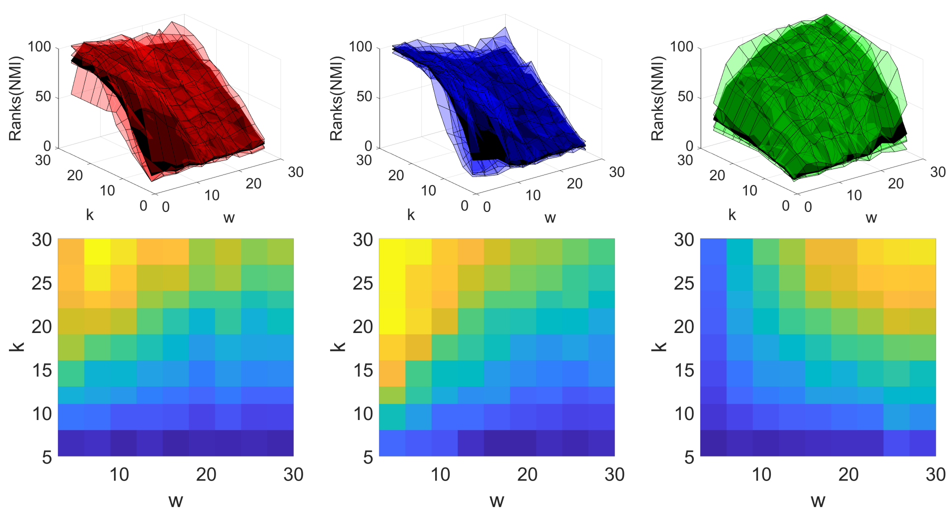


Fig. 3. Results for the three chosen metrics against the plane spanned by k and ws, for the five videos. The three clustering methods and the three baselines are shown as surfaces of different colours.

Table 2. Comparison of offline (ALL) versus online ($\gamma = 0$) clustering (non-constrained) for the five videos for the three metrics. m is the value of the respective metric. k and ws are the best parameter values for the non-constrained clustering methods.

	NMI					ARI					HCA				
	ALL		$\gamma = 0$			ALL		$\gamma = 0$			ALL		$\gamma = 0$		
Video	k	m	k	ws	m	k	m	k	ws	m	k	m	k	ws	m
Koi	30	0.48	11	3	0.67	30	0.19	11	3	0.52	13	0.35	11	3	0.64
Pigeons (Ground)	30	0.33	27	6	0.44	24	0.11	8	3	0.23	11	0.28	8	3	0.40
Pigeons (Kerb)	30	0.39	27	3	0.58	27	0.18	27	3	0.40	16	0.35	27	3	0.51
Pigeons (Square)	27	0.54	30	3	0.65	27	0.32	30	3	0.42	22	0.45	30	3	0.54
Pigs	30	0.40	30	6	0.55	16	0.17	30	6	0.39	16	0.31	30	6	0.50

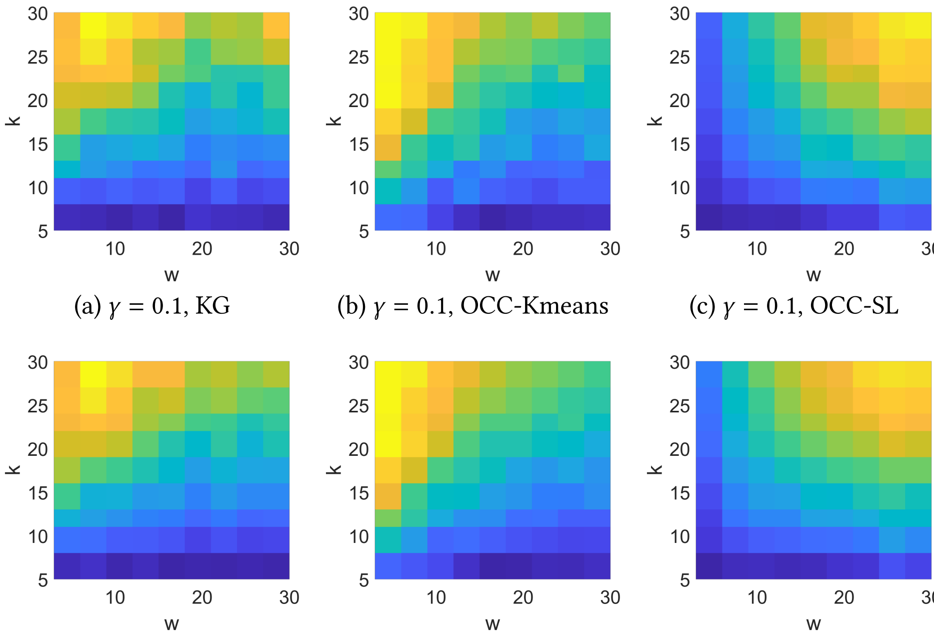


(a) KG (b) OCC-Kmeans (c) OCC-SL

Fig. 4. NMI ranks for the three clustering methods. Each plot in the top row contains five transparent surfaces of NMI ranks corresponding to the 5 videos, and one solid surface, which is the average rank. The heat map corresponds to the average rank surface.

(k, ws) will always receive value of 100, regardless of the video, and the worst combination will receive rank 1. Subsequently, we plotted the ranks for the three methods. In each plot, we show transparent surfaces for the individual videos and a single solid colour as the average rank surface. Under each 3D plot, we show the flat surface where yellow corresponds to large values (preferable) and dark blue, to low values.

The figure shows that KG and OCC-Kmeans are not as affected by window size as OCC-SL. For the former, a lower value of ws is preferred, but a mid value is also acceptable. For the latter, we can see that the method performs best when both the values of k and ws are high. For the value of k, on the other hand, one can see a clear correlation between higher accuracy and higher initial value of clusters.



(a) $\gamma = 0.1$, KG (b) $\gamma = 0.1$, OCC-Kmeans (c) $\gamma = 0.1$, OCC-SL

(d) $\gamma = 0.9$, KG (e) $\gamma = 0.9$, OCC-Kmeans (f) $\gamma = 0.9$, OCC-SL

Fig. 5. Heat maps for the NMI ranks for the three clustering methods for two different values of γ.

4.3 Effect of the number of constraints

Here we ask the question whether the number of constraints (represented by proportion γ) has an effect on the optimal combination of parameters (k, ws). Observing that the three metrics behave similarly, we chose to use only NMI for this part of the study. To answer the question, we plot a heatmap of NMI in the space spanned by (k, ws), for values of γ 0.1 and 0.9. Fig. 5 shows the results.

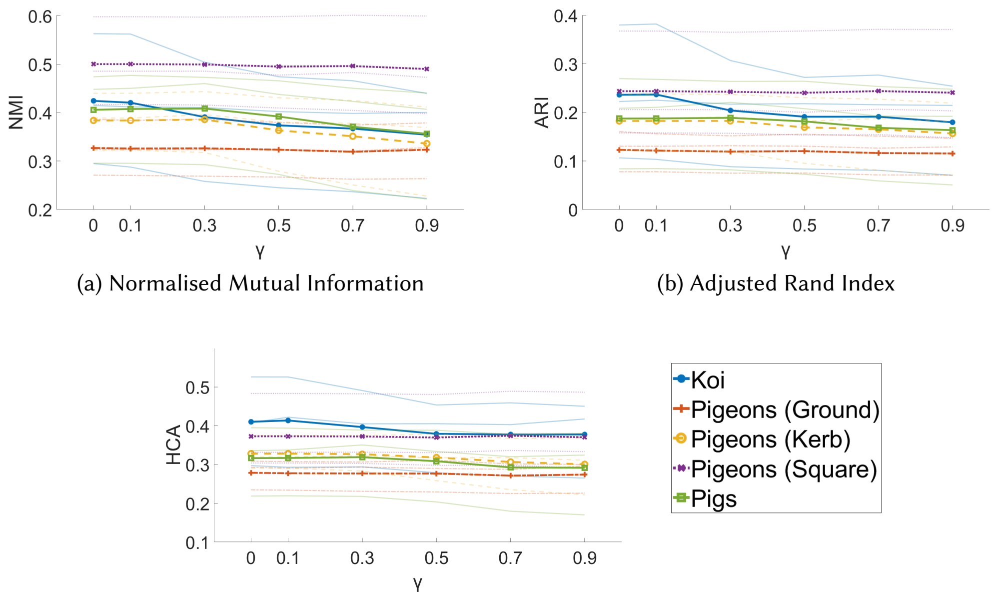
Each heat map in the figure is calculated by averaging the NMI ranks of the 5 videos for the respective γ. It can be seen that there is minimal difference between the plots for the different values of γ, which suggests that the number of constraints does not affect our recommendation of the optimal parameter combination.

We observed that in our experiment, the number of constraints did not have much effect on the quality of the clustering method. To visualise this statement, we include Fig. 6. The expected tendency was that increasing the number of constraints will improve the values of the metrics. Interestingly, our metrics show that increasing the number of constraints has the opposite, or very little, effect. More advanced future online constrained clustering methods may make better use of the constraints, which may change the parameter landscape.

5 CONCLUSION

Here we examine online constrained clustering applied to re-identify animals from video. Our experiment compared three online CC methods: KG, OCC-Kmeans, and OCC-SL. We are interested in how parameter choices affect the clustering results.

Curiously, we found that the number of constraints (γ) does not have an impact on the result. Contrary to expectation, the clustering accuracy did not improve with increasing the number of constraints for any of the videos, methods or metrics. This points at the need for new, more



(a) Normalised Mutual Information

(b) Adjusted Rand Index

(c) Classification Accuracy

Fig. 6. Plots of the metrics for each value of γ per method (opaque) and the method averages (bold).

advanced and effective, online constrained clustering methods. On the other hand, a combination of the number of clusters k and window size ws jointly affect the outcome. The preferred combination of values was found to be small ws and large k for KG and OCC-Kmeans, and large ws with large k for OCC-SL.

Our comparison of offline and our online methods with no constraints shows that adding a time component to the clustering has a positive effect on the results, even if the proportion of constraints does not.

The experimental results show that, in most cases, the resulting number of clusters in the optimum combinations of parameters does not match closely the true number of clusters in the ground truth. This shows that the methods are not sufficiently good at splitting and merging clusters.

Our future line of research includes modifications to the methods examined in this study or developing new methods that will be adequately equipped to adjust the number of clusters. We will be looking at online constrained Gaussian mixture modelling as well as more robust versions of hierarchical online constrained clustering.

ACKNOWLEDGMENTS

This work is supported by the UKRI Centre for Doctoral Training in Artificial Intelligence, Machine Learning and Advanced Computing (AIMLAC), funded by grant EP/S023992/1.

REFERENCES

[1] Francesco Alesiani, Gulcin Ermis, and Konstantinos Gkiotsalitis. 2022. Constrained clustering for the capacitated vehicle routing problem (cc-cvrp). *Applied Artificial Intelligence* 36, 1 (2022), 1995658. https://doi.org/10.1080/08839514.2021.1995658

[2] Amin Allahyar, Hadi Sadoghi Yazdi, and Ahad Harati. 2015. Constrained semi-supervised growing self-organizing map. *Neurocomputing* 147 (2015), 456–471. https://doi.org/10.1016/j.neucom.2014.06.039

[3] Samadhi Wickrama Arachchilage and Ebroul Izquierdo. 2020. Adaptive aggregated tracklet linking for multi-face tracking. In *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 1366–1370. https://doi.org/10.1109/ICIP40778.2020.9190823

[4] Michele Borassi, Alessandro Epasto, Silvio Lattanzi, Sergei Vassilvitskii, and Morteza Zadimoghaddam. 2020. Sliding window algorithms for k-clustering problems. *Advances in Neural Information Processing Systems* 33 (2020), 8716–8727. https://doi.org/10.5555/3495724.3496455

[5] Ye Cai and HaiYang Gan. 2019. An online face clustering algorithm for face monitoring and retrieval in real-time videos. In *IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*. IEEE, 825–830. https://doi.org/10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00122

[6] Feng Cao, Martin Estert, Weining Qian, and Aoying Zhou. 2006. Density-based clustering over an evolving data stream with noise. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. SIAM, 328–339. https://doi.org/10.1137/1.9781611972764.29

[7] Xiaochun Cao, Changqing Zhang, Chengju Zhou, Huazhu Fu, and Hassan Foroosh. 2015. Constrained multi-view video face clustering. *IEEE Transactions on Image Processing* 24, 11 (2015), 4381–4393. https://doi.org/10.1109/TIP.2015.2463223

[8] Carmela Comito, Clara Pizzuti, and Nicola Procopio. 2016. Online clustering for topic detection in social data streams. In *IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*. IEEE, 362–369. https://doi.org/10.1109/ICTAI.2016.0062

[9] Thiago F Covões, Eduardo R Hruschka, and Joydeep Ghosh. 2012. Competitive learning with pairwise constraints. *IEEE Transactions on Neural Networks and Learning Systems* 24, 1 (2012), 164–169. https://doi.org/10.1109/TNNLS.2012.2227064

[10] Ming Du and Rama Chellappa. 2012. Face association across unconstrained video frames using conditional random fields. In *12th European Conference on Computer Vision (ECCV)*. Springer, 167–180. https://doi.org/10.1007/978-3-642-33786-4_13

[11] Germán González-Almagro, Daniel Peralta, Eli De Poorter, José-Ramón Cano, and Salvador García. 2023. Semi-supervised constrained clustering: an in-depth overview, ranked taxonomy and future research directions. *arXiv preprint arXiv:2303.00522* (2023). https://doi.org/10.48550/arXiv.2303.00522

[12] Maria Halkidi, Myra Spiliopoulou, and Aikaterini Pavlou. 2012. A semi-supervised incremental clustering algorithm for streaming data. In *Advances in Knowledge Discovery and Data Mining: 16th Pacific-Asia Conference (PAKDD 2012)*. Springer, 578–590. https://doi.org/10.1007/978-3-642-30217-6_48

[13] Lawrence Hubert and Phipps Arabie. 1985. Comparing partitions. *Journal of Classification* 2 (1985), 193–218. https://doi.org/10.1007/BF01908075

[14] Hurieh Khalajzadeh, Mohammad Mansouri, and Mohammad Teshnehlab. 2014. Face recognition using convolutional neural network and simple logistic classifier. In *Soft Computing in Industrial Applications: Proceedings of the 17th Online World Conference on Soft Computing in Industrial Applications*. Springer, 197–207. https://doi.org/10.1007/978-3-319-00930-8_18

[15] Harold W Kuhn. 1955. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* 2, 1-2 (1955), 83–97. https://doi.org/10.1002/nav.3800020109

[16] Prakhar Kulshreshtha and Tanaya Guha. 2018. An online algorithm for constrained face clustering in videos. In *25th IEEE International Conference on Image Processing (ICIP)*. IEEE, 2670–2674. https://doi.org/10.1109/ICIP.2018.8451343

[17] Ludmila I Kuncheva, Francis Williams, Samuel L Hennessey, and Juan J Rodríguez. 2022. A benchmark database for animal re-identification and tracking. In *5th IEEE International Conference on Image Processing Applications and Systems (IPAS)*. IEEE, 1–6. https://doi.org/10.1109/IPAS55744.2022.10052988

[18] Edo Liberty, Ram Sriharsha, and Maxim Sviridenko. 2016. An algorithm for online k-means clustering. In *Proceedings of the Eighteenth Workshop on Algorithm Engineering and Experiments (ALENEX)*. SIAM, 81–89. https://doi.org/10.1137/1.9781611974317.7

[19] Claudia Malzer and Marcus Baum. 2021. Constraint-based hierarchical cluster selection in automotive radar data. *Sensors* 21, 10 (2021), 3410. https://doi.org/10.3390/s21103410

[20] Igor Melnykov and Volodymyr Melnykov. 2020. A note on the formal implementation of the k-means algorithm with hard positive and negative constraints. *Journal of Classification* 37, 3 (2020), 789–809. https://doi.org/10.1007/s00357-019-09349-x

[21] Adway Mitra, Soma Biswas, and Chiranjib Bhattacharyya. 2016. Bayesian modeling of temporal coherence in videos for entity discovery and summarization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 3 (2016), 430–443. https://doi.org/10.1109/TPAMI.2016.2557785

[22] David Montero, Naiara Aginako, Basilio Sierra, and Marcos Nieto. 2022. Efficient large-scale face clustering using an online mixture of gaussians. *Engineering Applications of Artificial Intelligence* 114 (2022), 105079. https://doi.org/10.1016/j.engappai.2022.105079

[23] Eshed Ohn-Bar and Mohan Manubhai Trivedi. 2015. Learning to detect vehicles by clustering appearance patterns. *IEEE Transactions on Intelligent Transportation Systems* 16, 5 (2015), 2511–2521. https://doi.org/10.1109/TITS.2015.2409889

[24] Puguh Budi Prakoso and Yuslena Sari. 2019. Vehicle detection using background subtraction and clustering algorithms. *Telkomnika* 17, 3 (2019). https://doi.org/10.12928/TELKOMNIKA.v17i3.10144

[25] Marian-Andrei Rizoiu, Julien Velcin, and Stéphane Lallich. 2014. How to use temporal-driven constrained clustering to detect typical evolutions. *International Journal on Artificial Intelligence Tools* 23, 04 (2014), 1460013. https://doi.org/10.1142/S0218213014600136

[26] Carlos Ruiz, Ernestina Menasalvas, and Myra Spiliopoulou. 2009. C-denstream: Using domain knowledge on a data stream. In *Discovery Science: 12th International Conference (DS 2009)*. Springer, 287–301. https://doi.org/10.1007/978-3-642-04747-3_23

[27] Alexander Strehl and Joydeep Ghosh. 2002. Cluster ensembles—a knowledge reuse framework for combining multiple partitions. *Journal of Machine Learning Research* 3, Dec (2002), 583–617. https://doi.org/10.1162/153244303321897735

[28] Makarand Tapaswi, Marc T Law, and Sanja Fidler. 2019. Video face clustering with unknown number of clusters. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 5027–5036. https://doi.org/10.1109/ICCV.2019.00513

[29] Jesper E Van Engelen and Holger H Hoos. 2020. A survey on semi-supervised learning. *Machine learning* 109, 2 (2020), 373–440. https://doi.org/10.1007/s10994-019-05855-6

[30] Nguyen Xuan Vinh, Julien Epps, and James Bailey. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary?. In *Proceedings of the 26th Annual International Conference on Machine Learning*. 1073–1080. https://doi.org/10.1145/1553374.1553511

[31] Kiri Wagstaff, Claire Cardie, Seth Rogers, and Stefan Schrödl. 2001. Constrained k-means clustering with background knowledge. In *Proceedings of the Eighteenth International Conference on Machine Learning*, Vol. 1. 577–584. https://doi.org/10.5555/645530.655669

[32] Wenchao Xiao, Yan Yang, Hongjun Wang, Tianrui Li, and Huanlai Xing. 2016. Semi-supervised hierarchical clustering ensemble and its application. *Neurocomputing* 173 (2016), 1362–1376. https://doi.org/10.1016/j.neucom.2015.09.009

[33] Weikai Yang, Xiting Wang, Jie Lu, Wenwen Dou, and Shixia Liu. 2020. Interactive steering of hierarchical clustering. *IEEE Transactions on Visualization and Computer Graphics* 27, 10 (2020), 3953–3967. https://doi.org/10.1109/TVCG.2020.2995100

[34] Peng Yao, Huaqiang Wu, Bin Gao, Sukru Burc Eryilmaz, Xueyao Huang, Wenqiang Zhang, Qingtian Zhang, Ning Deng, Luping Shi, H-S Philip Wong, et al. 2017. Face classification using electronic synapses. *Nature Communications* 8, 1 (2017), 15199. https://doi.org/10.1038/ncomms15199

[35] Jinfeng Yi, Lijun Zhang, Tianbao Yang, Wei Liu, and Jun Wang. 2015. An efficient semi-supervised clustering algorithm with sequential constraints. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1405–1414. https://doi.org/10.1145/2783258.2783389