Data Stream Mining Based on Ant Colony Behaviour

Shengxiang Yang

Centre for Computational Intelligence (CCI)
School of Computer Science and Informatics
De Montfort University, Leicester LE1 9BH, UK
Email: syang@dmu.ac.uk
http://www.tech.dmu.ac.uk/~syang

Outline of the Talk

- Introduction to data stream
  - Concept drift and evolution
- Clustering for data stream
  - Ant Colony Stream Clustering (ACSC)
  - Multi-density Stream Clustering (MDSC)
- Classification in dynamic streams
  - Clustering and One Class Ensemble Learning (COCEL)
- Summary

Data Stream Formally

- Stream $S = [i^t]_{t=0}^{\infty}$, where $i^t = (x^t, y^t)$
- Point x in d dimensions, $x^t = \{v_1, \ldots, v_d\}$, describes concept y at time t where $y \in Y$
- Using probability notation: $P(y^t|x^t)$

Data Stream Mining

- Given a data stream S, extract information from S

- Challenges:
  - **Time** Constraints
    - Points should be processed in a single pass
  - **Memory** Constraints
    - Stream potentially infinite, memory finite
  - **Dynamic**
    - Characteristics of data can **change** in unforeseen ways

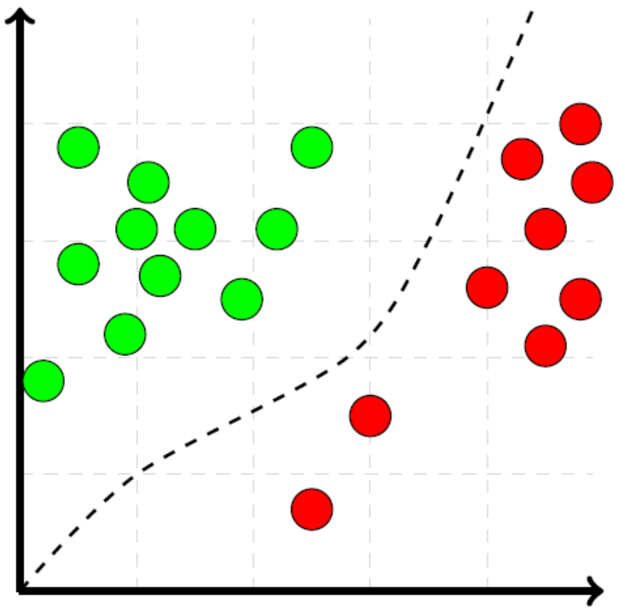
Types of Change in Data Streams

- Concept Drift
  - Virtual drift: Change in $P(x)$
  - Real drift: Change in $P(y|x)$

- Concept Evolution
  - New concepts appear in stream, $y^t \notin Y$

Concept Drift: Virtual vs Real

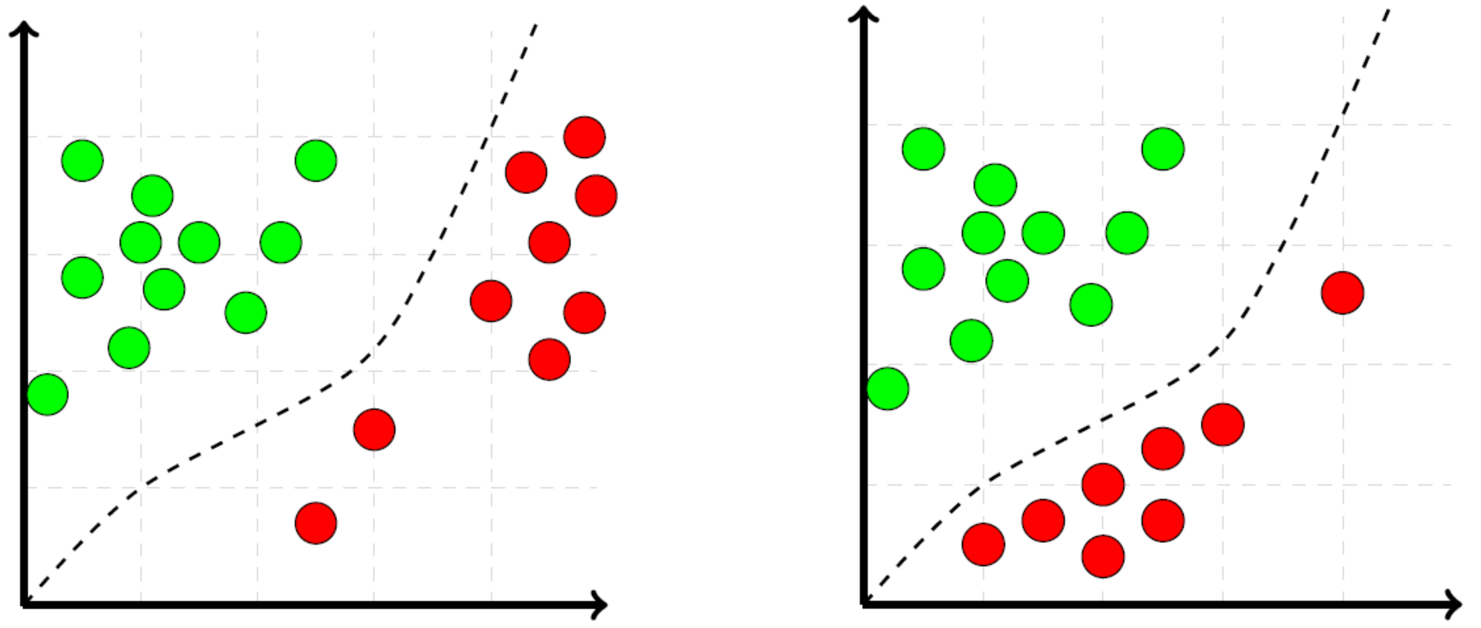
- Concept before change



(a) Raw Data

Virtual Concept Drift

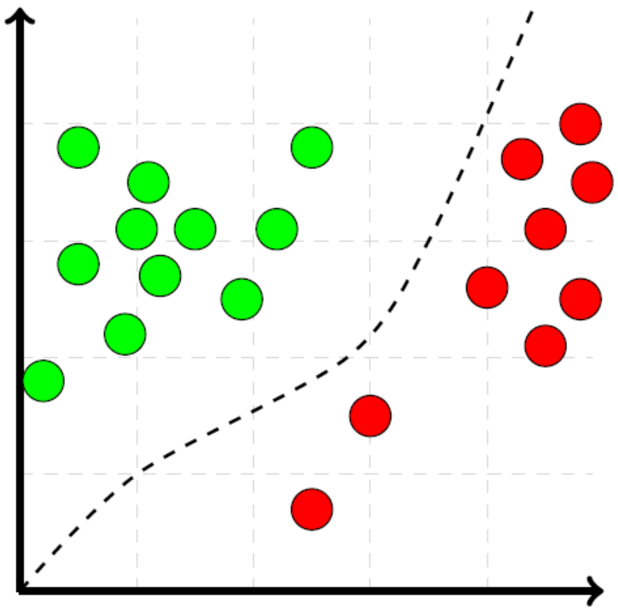
- Change in X (i.e., $P(x)$ change) but no change in decision boundary

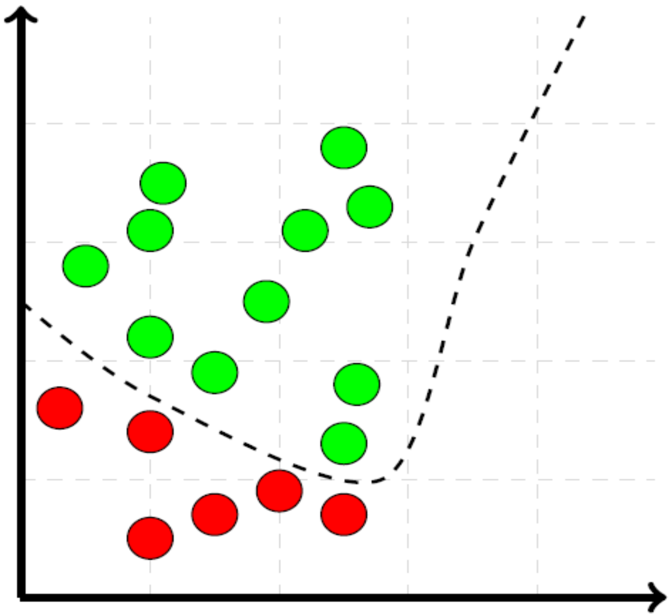


(a) Raw Data (b) Virtual Drift

Real Concept Drift

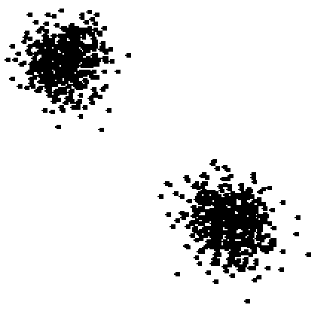
- Change in decision boundary, i.e., $P(y|x)$ change


(a) Raw Data

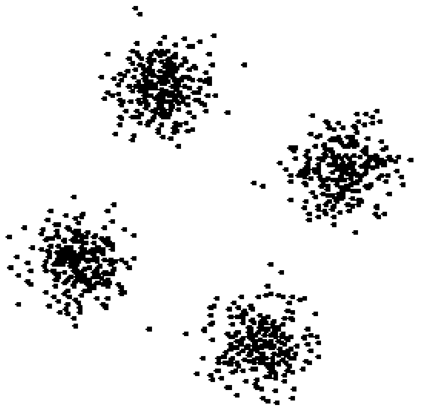

(c) Real Drift

Concept Drift Examples

- Virtual drift: Change in $P(x)$
  - Change in source distribution, decision boundary unaffected

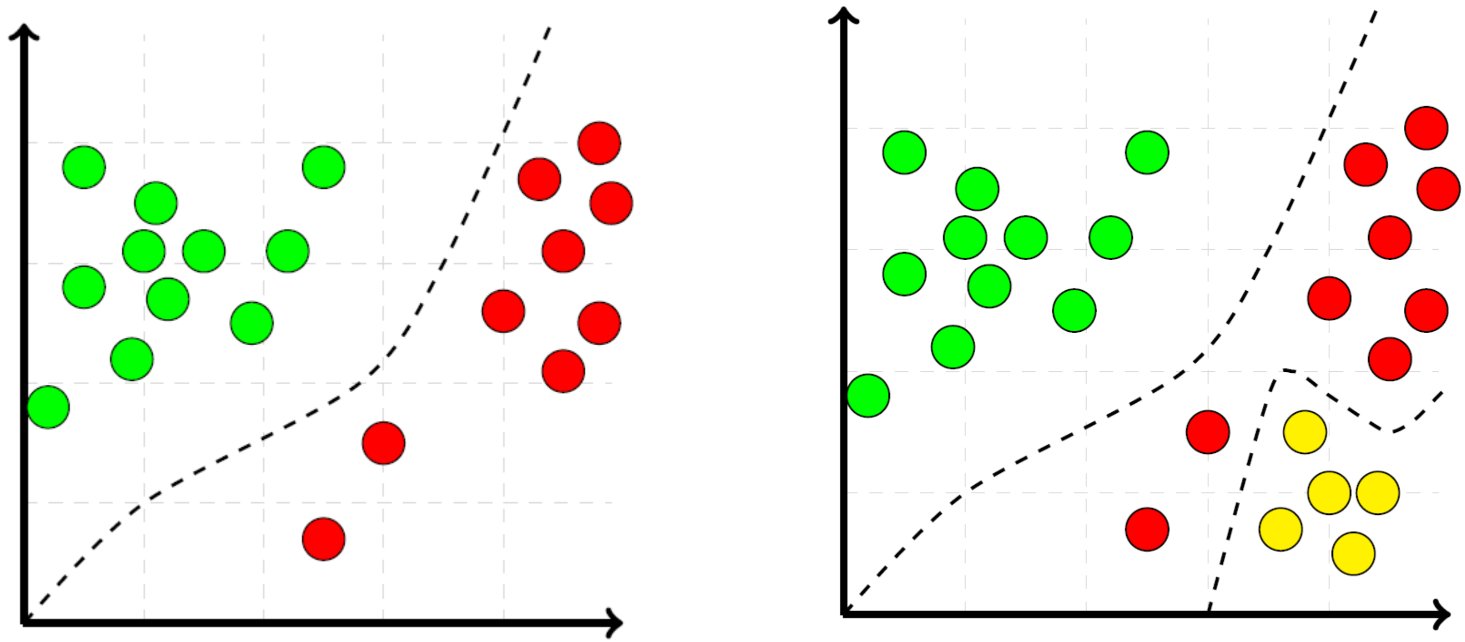


- Real drift: Change in $P(y|x)$
  - Decision boundary changes



Concept Evolution

- New class appears after time t



(a) Raw Data

(b) Concept Evolution

Detecting Changes

- Supervised methods
  - Assuming labels for incoming points are available and inexpensive to collect
- Unsupervised methods
  - Labels not immediately available or labels are expensive
  - One possible way is to identify clusters in the stream and **track** these clusters over-time to detect underlying change

Clustering with ACO

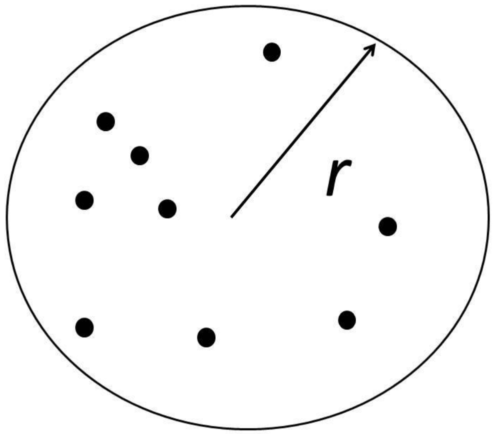
- Clustering problem framed as an optimisation problem
- Usually, cluster centres are optimised and points clustered using k-means
  - Useful in static clustering (Nikham,2010; Shelokar et al., 2004)
- Problematic in stream clustering:
  - How many centres to find? **K can change**...
  - Iterative, population-based searching can be **slow**
- Ant Colony Stream Clustering (**ACSC**) (Fahy et al., 2019)
  - Density based clustering
  - Nest building and nest sorting behaviour of ants

C. Fahy, S. Yang, M. Gongora. Ant colony stream clustering: A fast density clustering algorithm for dynamic data streams. IEEE Transactions on Cybernetics, 49(6): 2215-2228, 2019

Density Based Clustering

- Clusters identified as areas of high density separated by areas of low density
  - K doesn't need to be specified

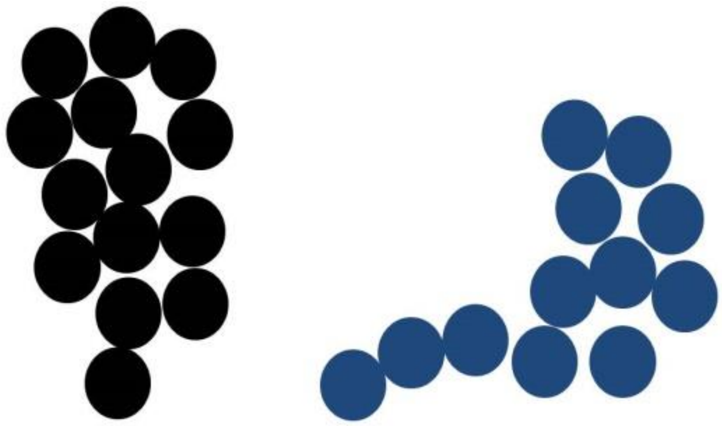
- Micro-clusters
  - **Summarise** similar points



Micro-cluster summarises points

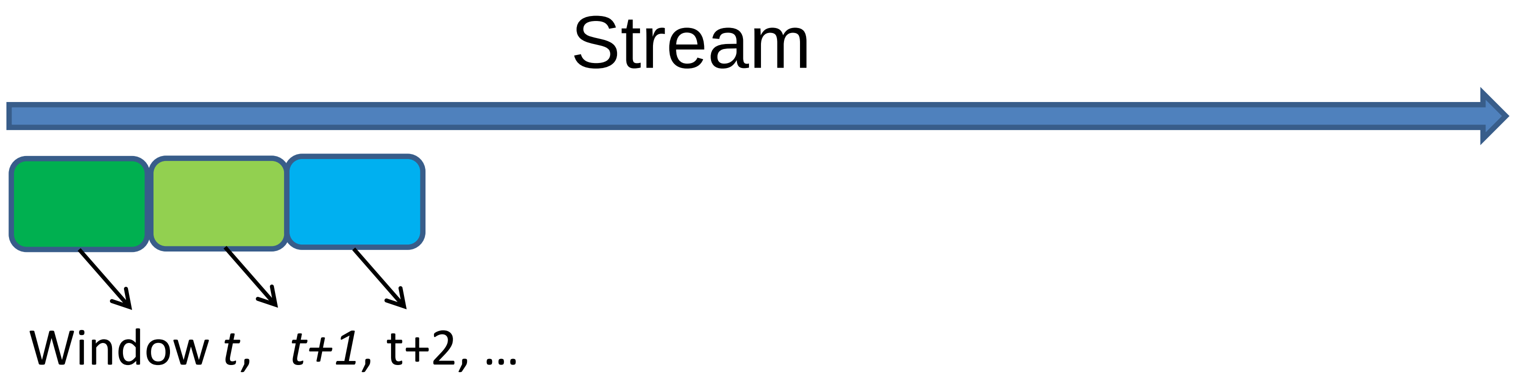
Density Based Clustering

- Two micro-clusters are ‘connected’ if the distance from their centres is less than ε
- Connected micro-clusters form the cluster



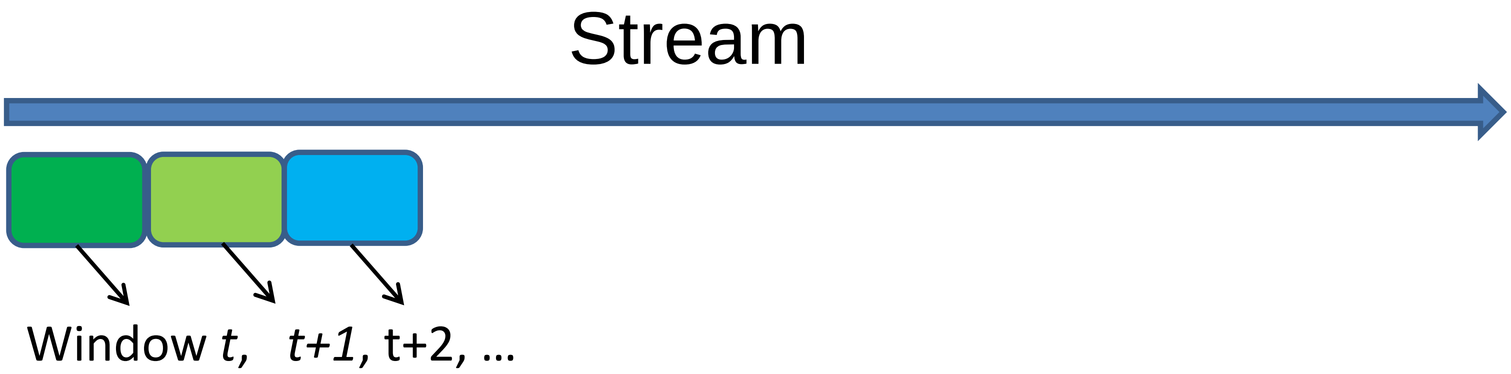
Two clusters composed of micro-clusters

ACSC Overview



- Read stream in **windows**
- **Cluster** each window
- **Summarise** each window

ACSC Overview

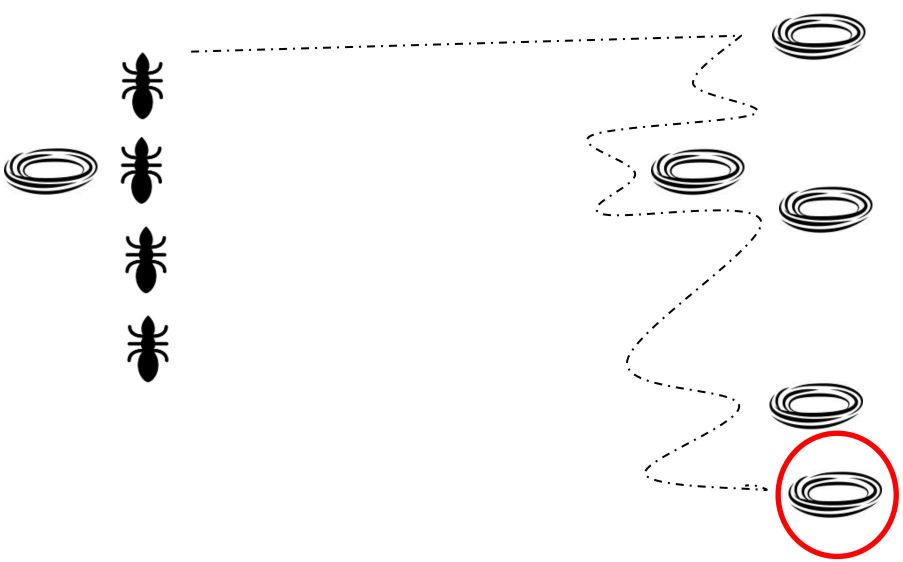


- Two steps to clustering:
    1) Initial clusters identified in a single pass of the window – **nest building**
    2) Initial clusters are refined – **nest sorting**

Nest Building

- Incoming stream ➡ Read Window
- Each point is an ‘ant’ ➡ ants form nests with similar ants
- First ant forms first nest

- Subsequent ants can join existing nest or start new nest



$$Sim(a,k) = \frac{\sum_{j=1}^{nComp} dist(a,k_j)}{nComp}$$

$Sim(a,k) \leq \varepsilon$

Join Nest

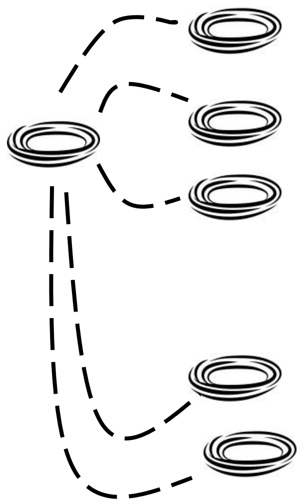
$Sim(a,k) > \varepsilon$

New Nest!

Nest Building

- Similarity score with each nest is recorded: **pheromone trails**
- Pheromone trail between nests *a* and *b* is the average similarity of each ant in *a* with nest *b*:

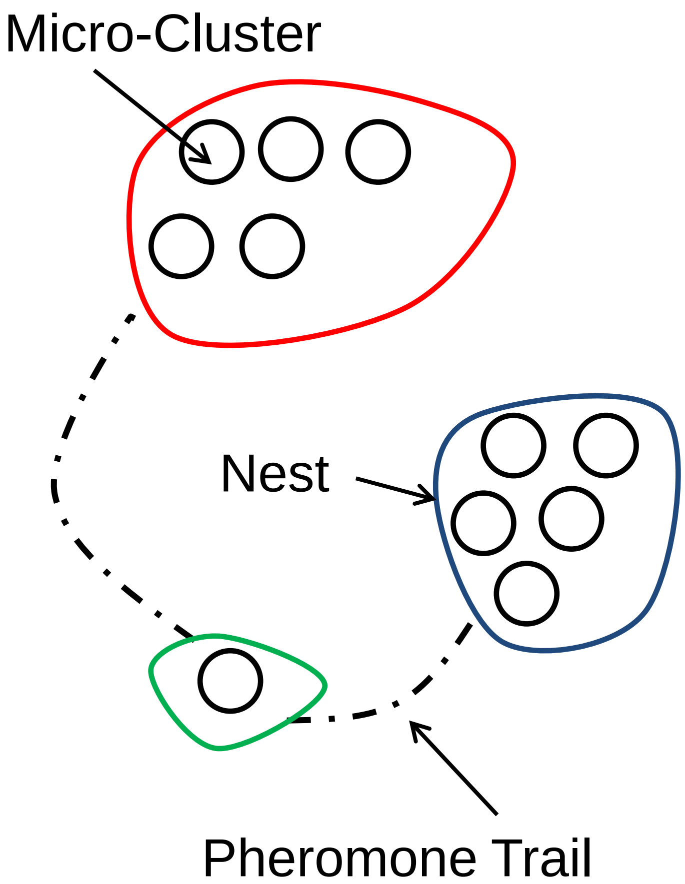
$$ph(a,b) = \frac{1}{n}\sum_{i=1}^{n} Sim(a_i, b)$$



- At the end of this step, a set of Nests and similarity between each pair of nests

$$\begin{bmatrix} ph(nest_1, nest_1) & \cdots & ph(nest_1, nest_n) \\ \vdots & \ddots & \vdots \\ ph(nest_n, nest_1) & \cdots & ph(nest_n, nest_n) \end{bmatrix}$$

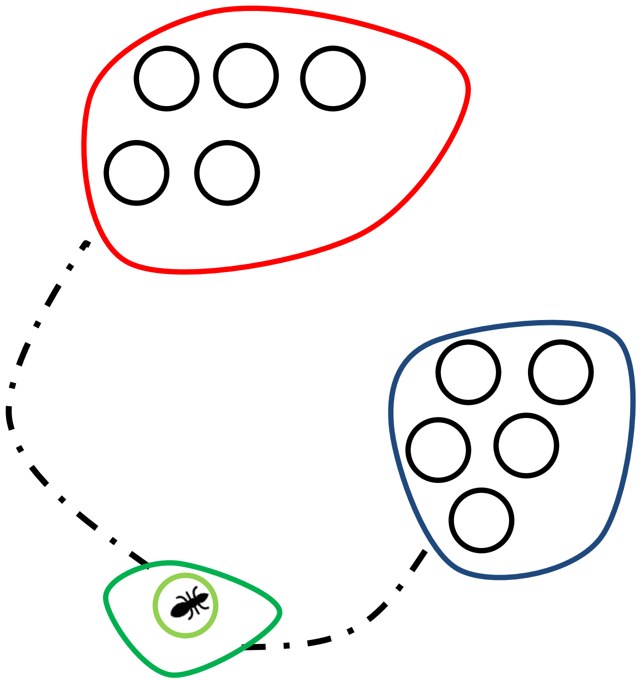
Nest Sorting



- Points in each nest are merged to form **micro-clusters**
- Based on observed sorting behaviour of ants: the **pick-and-drop** model (Lumar and Faieta, 1994)
- Ants pick-up isolated items and drop in locations where similar items are present.
- Biologically: corpses, eggs etc.
- Here, micro-clusters…

E. Lumar, B. Faieta. Diversity and adaptation in populations of clustering ants. Proc. 3rd Int. Conf. on Simulation of Adaptive Behavior: From Animals to Animats, vol. 3, pp. 489–508, 1994

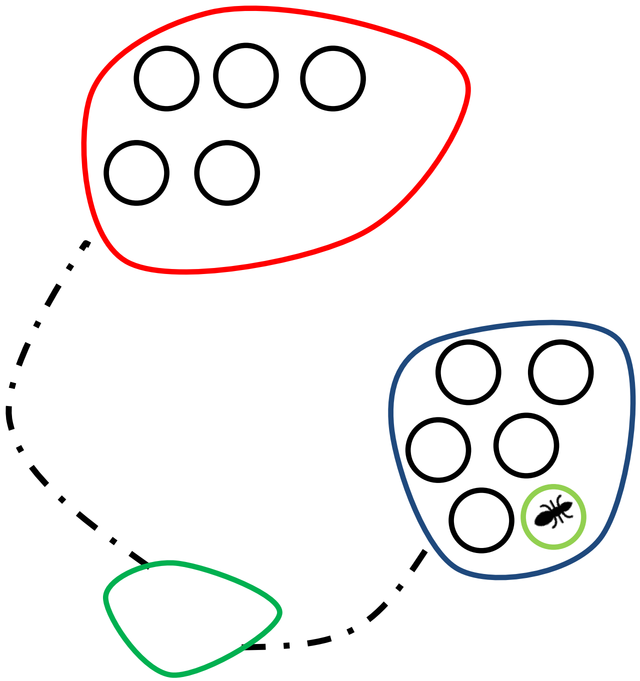
Nest Sorting



- Each nest is assigned a sorting ant
- Ant picks up a micro-cluster

$$P_{pick} = 1 - \frac{numConnectedMCs}{Samples}$$

Nest Sorting



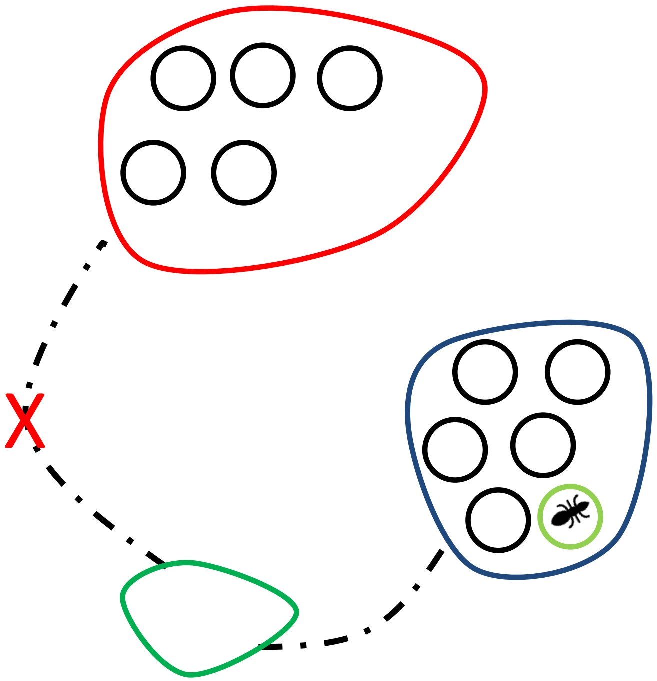
- Each nest is assigned a sorting ant
- Ant picks up a micro-cluster

$$P_{pick} = 1 - \frac{numConnectedMCs}{Samples}$$

- If pick is successful, ant moves to similar nest and attempts to drop in new nest:

$$P_{drop} = \frac{numConnectedMCs}{Samples}$$

Nest Sorting



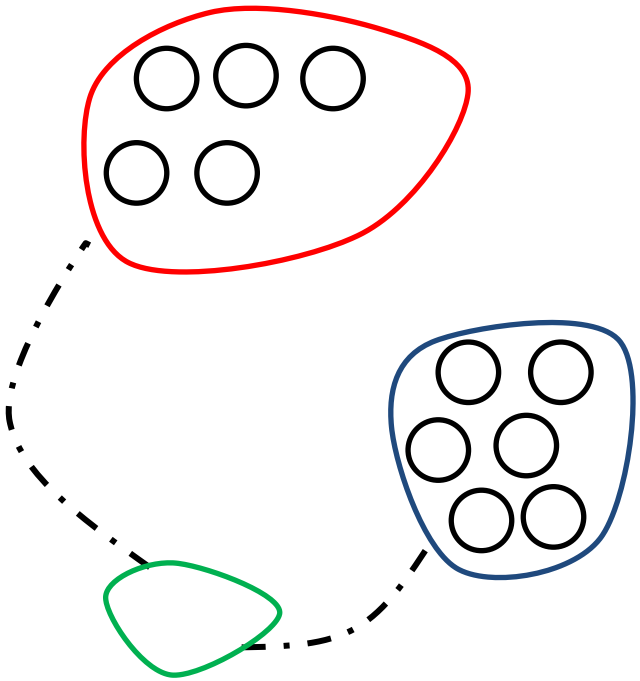
- Each nest is assigned a sorting ant
- Ant picks up a micro-cluster

$$P_{pick} = 1 - \frac{numConnectedMCs}{Samples}$$

- If pick is successful, ant moves to **similar** nest and attempts to drop in new nest:

$$P_{drop} = \frac{numConnectedMCs}{Samples}$$

Nest Sorting



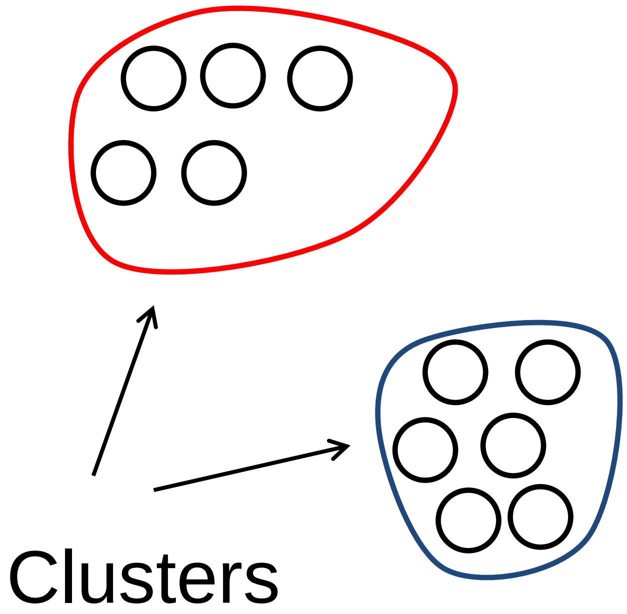
- Each nest is assigned a sorting ant
- Ant picks up a micro-cluster

$$P_{pick} = 1 - \frac{numConnectedMCs}{Samples}$$

- If pick is successful, ant moves to similar nest and attempts to drop in new nest:

$$P_{drop} = \frac{numConnectedMCs}{Samples}$$

Nest Sorting



- Non empty nests are clusters
- Clusters are summarised by their micro-clusters (number of micro-clusters and their centres)
- Summaries stored off-line and next window evaluated
- New clusters or a change in micro-cluster centres signal change in stream…

ACSC Comparative Results – Quality

- Compared with peer stream-clustering algorithms
  - Performance: Cluster Purity, F1 Score, Rand Index

	DenStream			*CluStream*			*ClusTree*			*ACSC*		
	P	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*
1CDT	0.99	0.82	0.77	**1.0**	0.88	0.80	**1.0**	0.89	0.82	0.99(s-)	**0.99**(s+)	**0.99**(s+)
2CHT	0.43	0.27	0.53	0.24	0.23	0.55	0.22	0.24	**0.58**	**0.81**(s+)	**0.42**(s+)	0.55(s-)
4CR	**1.00**	0.67	0.71	**1.00**	0.89	0.89	**1.00**	0.89	0.89	0.99(s-)	**0.95**(s+)	**0.97**(s+)
4CE1CF	**0.99**	0.35	0.56	**0.99**	0.86	0.89	**0.99**	0.86	0.89	0.96(s-)	0.76(s-)	**0.90**(s+)
Network	**1.00**	0.80	0.81	0.35	0.13	0.36	0.36	0.16	0.3	**1.0**(=)	**0.95**(s+)	**0.95**(s+)
CoverType	**0.89**	0.10	0.51	0	0	0	0	0	0	0.88(s-)	**0.59**(s+)	**0.64**(s+)
Average	0.88	0.50	0.64	0.59	0.49	0.58	0.59	0.51	0.58	**0.93**	**0.77**	**0.83**

ACSC Comparative Results – Time

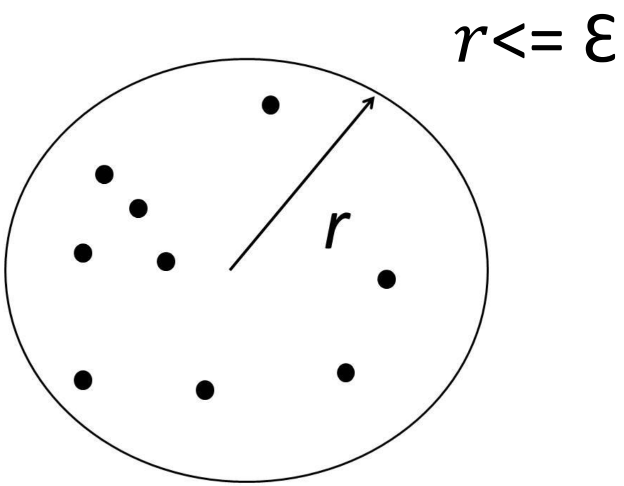
	DenStream		*CluStream*		*ClusTree*		*ACSC*	
	Total,	Window	Total,	Window	Total,	Window	Total,	Window
1CDT	05.74	0.38(0.06)	01.69	0.11(0.02)	01.22	0.07(0.01)	**0.71**(0.01)	**0.05**(0.02)
2CHT	05.61	0.37(0.05)	01.67	0.11 (0.02)	01.38	0.09 (0.02)	**0.62** (0.06)	**0.05**(0.02)
4CR	50.62	0.29(0.04)	11.78	0.09(0.01)	12.11	0.09(0.01)	**09.28**(0.1)	**0.06**(0.01)
4CE1CF	55.06	0.38(0.03)	14.64	0..08(0.01)	**12.96**	**0.08**(0.41)	16.85(0.3)	0.09(0.01)
Network	94.41	0.19(0.77)	106.21	0.22(0.18)	22.11	0.06(0.3)	**20.63**(0.3)	**0.04**(0.02)
CoverType	278.5	0.56(0.09)	26.62	0.04(0.02)*	22.07	0.03(0.02)*	**49.53**(1.07)	**0.08**(0.02)

* Did not return a clustering solution

- ACSC: Better performance and faster

ACSC Drawbacks

- Ɛ determines maximum radius of micro-cluster
- Manually tuned, very sensitive parameter
- Ɛ is **global** so restricts the algorithm to a **single level of density**



Micro-cluster with radius r

- Clusters not 'online'
- Windowing model used – behaviour of dynamic clusters cannot be tracked over time

Multi Density Stream Clustering (MDSC)

- MDSC extends ACSC concepts

	Ɛ Parameter	Clustering Process	Density
ACSC	Manually Tuned	Two-Phase: Online and Offline	Single density
MDSC	Adaptive	Single Phase: Online	Multi-density

C. Fahy, S. Yang. Finding and Tracking Multi-Density Clusters in Dynamic Data Streams. IEEE Transactions on Big Data, in press, 2019 (DOI: 10.1109/TBDATA.2019.2922969).

MDSC Comparative Results

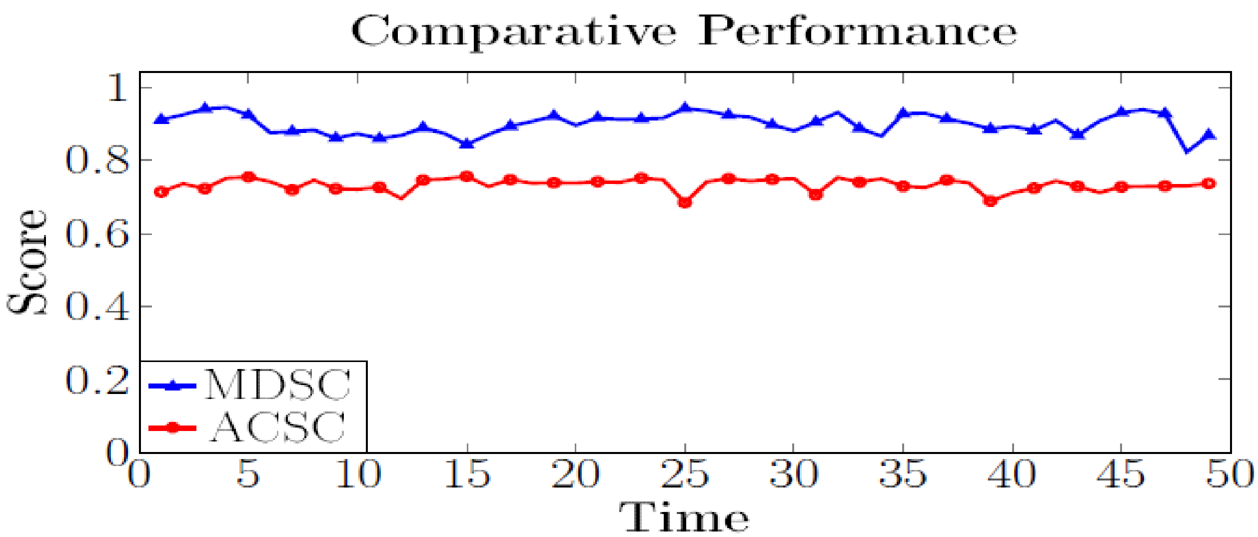
- Compared with ACSC and three other peer clustering algorithms on three metrics
  - Cluster Purity, F1 Score, Rand Index

	DenStream			*MuDi*			*CEDAS*			*ACSC*			*MDSC*		
	P	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*	*P*	*F*	*R*
Network	**1.00**	0.61	0.80	0.97	0.87	0.81	0.99	**0.95**	**0.96**	**1.00**	**0.95**	0.94	0.99(s-)	0.93(s-)	0.94(s-)
Forest	0.79	0.10	0.51	0.73	0.47	0.52	0.86	0.48	0.59	0.88	0.59	0.64	**0.89**(s+)	**0.61**(s+)	**0.66**(s+)
KeySroke	0.86	0.16	0.54	0.61	0.46	0.70	0.87	0.61	0.67	0.88	0.56	0.68	**0.88**(=)	**0.65**(s+)	**0.77**(s+)
COIL	0.00	0.00	0.00	0.84	0.67	0.64	0.50	0.17	0.23	0.86	0.76	0.74	**0.92**(s+)	**0.81**(s+)	**0.81**(s+)
2CSurr	0.88	0.22	0.51	0.90	0.76	0.67	**0.97**	0.61	0.61	**0.97**	0.62	0.60	**0.97**(=)	**0.89**(s+)	**0.80**(s+)
4CR	**1.00**	0.67	0.71	0.94	0.94	0.91	0.98	0.95	0.96	**1.00**	0.95	0.97	**1.00**(=)	**0.98**(s+)	**0.98**(s+)
20D	0.84	0.22	0.23	0.92	0.87	0.94	0.98	0.79	0.93	0.96	0.77	0.93	**0.99**(s+)	**0.94**(s+)	**0.97**(s+)
Average	0.76	0.2	0.47	0.84	0.72	0.74	0.87	0.65	0.7	0.93	0.74	0.78	**0.94**	**0.83**	**0.84**

- ACSC is faster but is restricted to a single level of density and requires careful manual tuning. MDSC is better for multi-density data

MDSC Comparison with ACSC

- Example Synthetic Stream: 2CR
  - Two classes in two dimensions
  - One class non-stationary
  - Two levels of density (multi-density clusters)

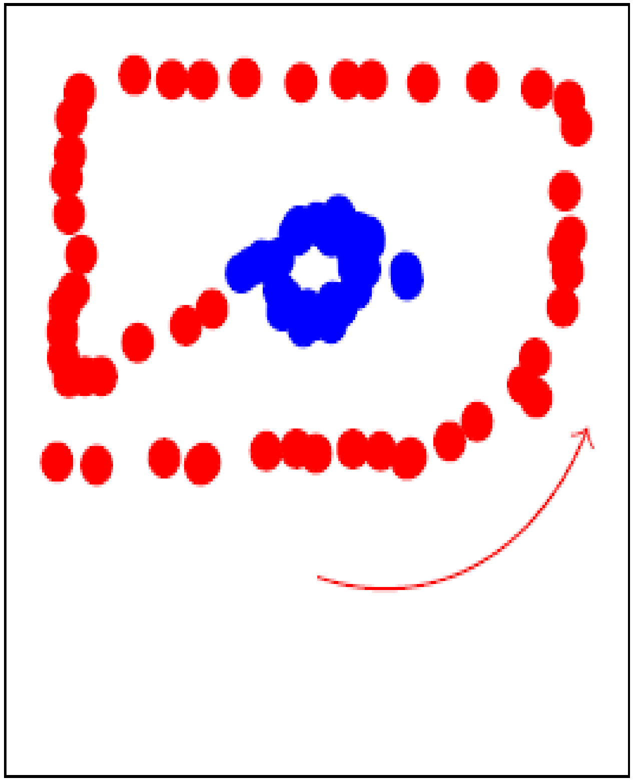


* Score is average of Purity, Rand Index and F1

- ACSC performance degrades in case of multi-density

MDSC Comparison with ACSC

- Cluster behaviour can be tracked and monitored with MDSC
- Blue cluster is stationary and red cluster drifts in the direction of arrow
- Centers of clusters are recorded every time-step and the drift is captured and tracked



Classification in Dynamic Streams

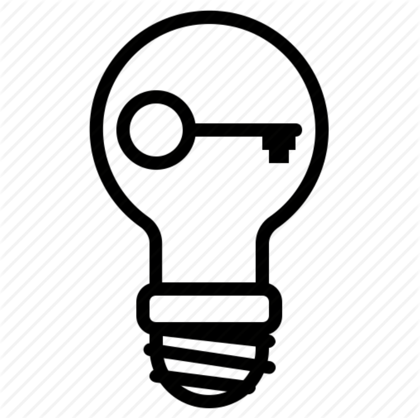
- Scarcity of labels
  - Most incoming points will not have labels
  - How to Train? Test?

- Clustering and classification ensemble

COCEL

- Clustering and One Class Ensemble Learning (COCEL)

- **Key Idea:**
  - Stream **Clustering** and an ensemble of **One Class Classifiers** with **Active Learning**



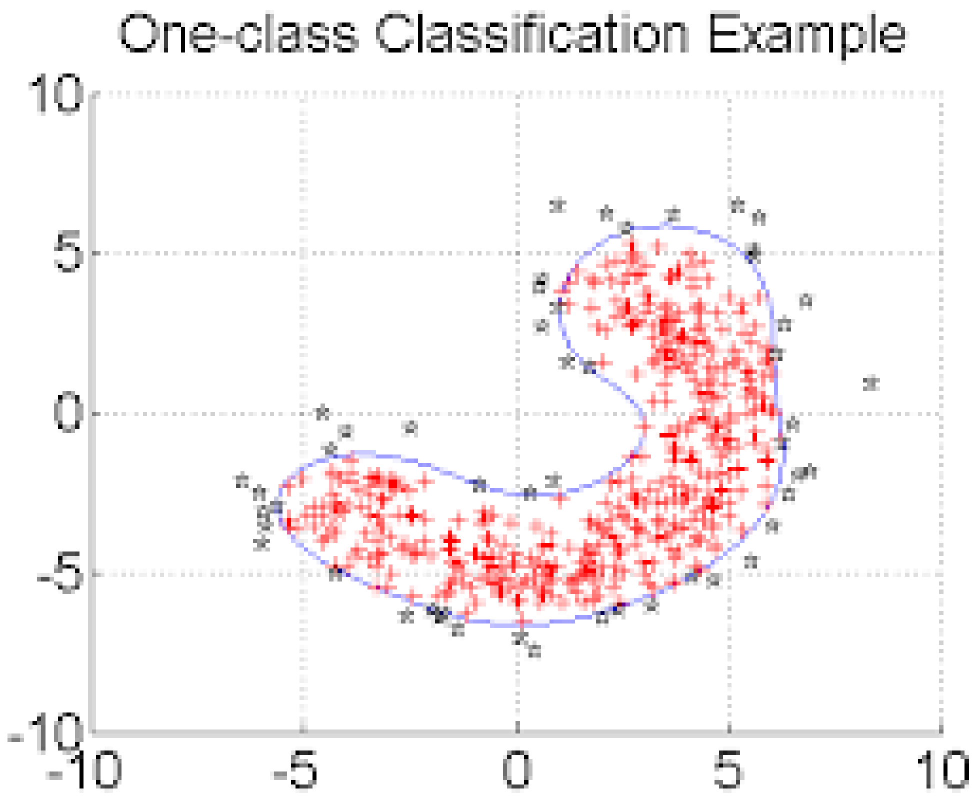
C. Fahy, S. Yang, M. Gongora. Classification in dynamic data streams with a scarcity of labels. IEEE Transactions on Knowledge and Data Engineering, submitted in March 2020.

One Class Classification

- Trained to recognise ONE particular class
- Examples:
  - Support vector domain description
  - Neural network auto-encoder
  - Principle Component Analysis (PCA)
  - Micro-classifiers
- Usually trained with only positive examples

One Class Classification

- Find a boundary around positive class



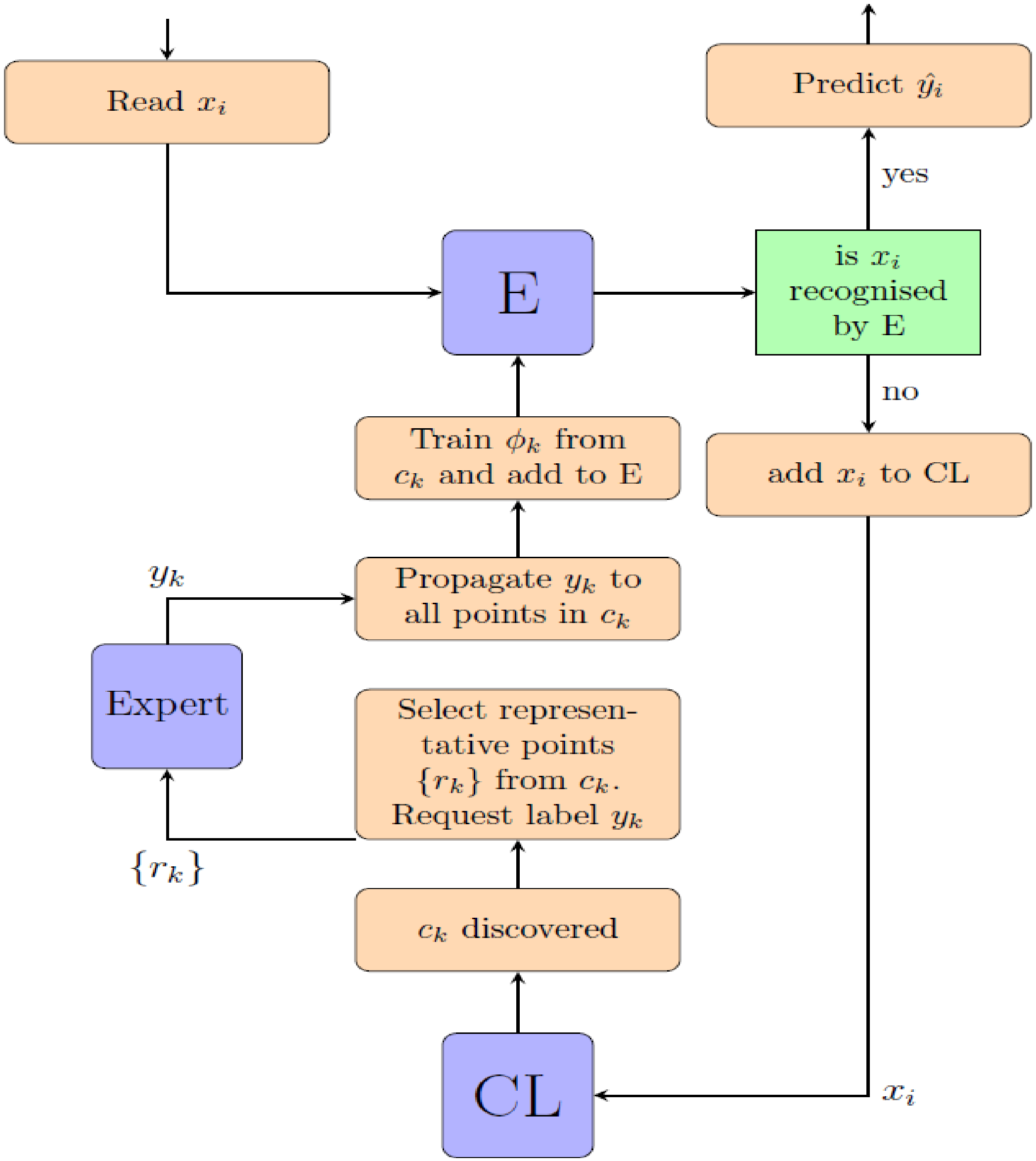
Support Vector Domain Description OCC

Active Learning

- Model **requests** a label for a specific sample
- Only give model samples that are useful
- Hugely reduces labelling costs

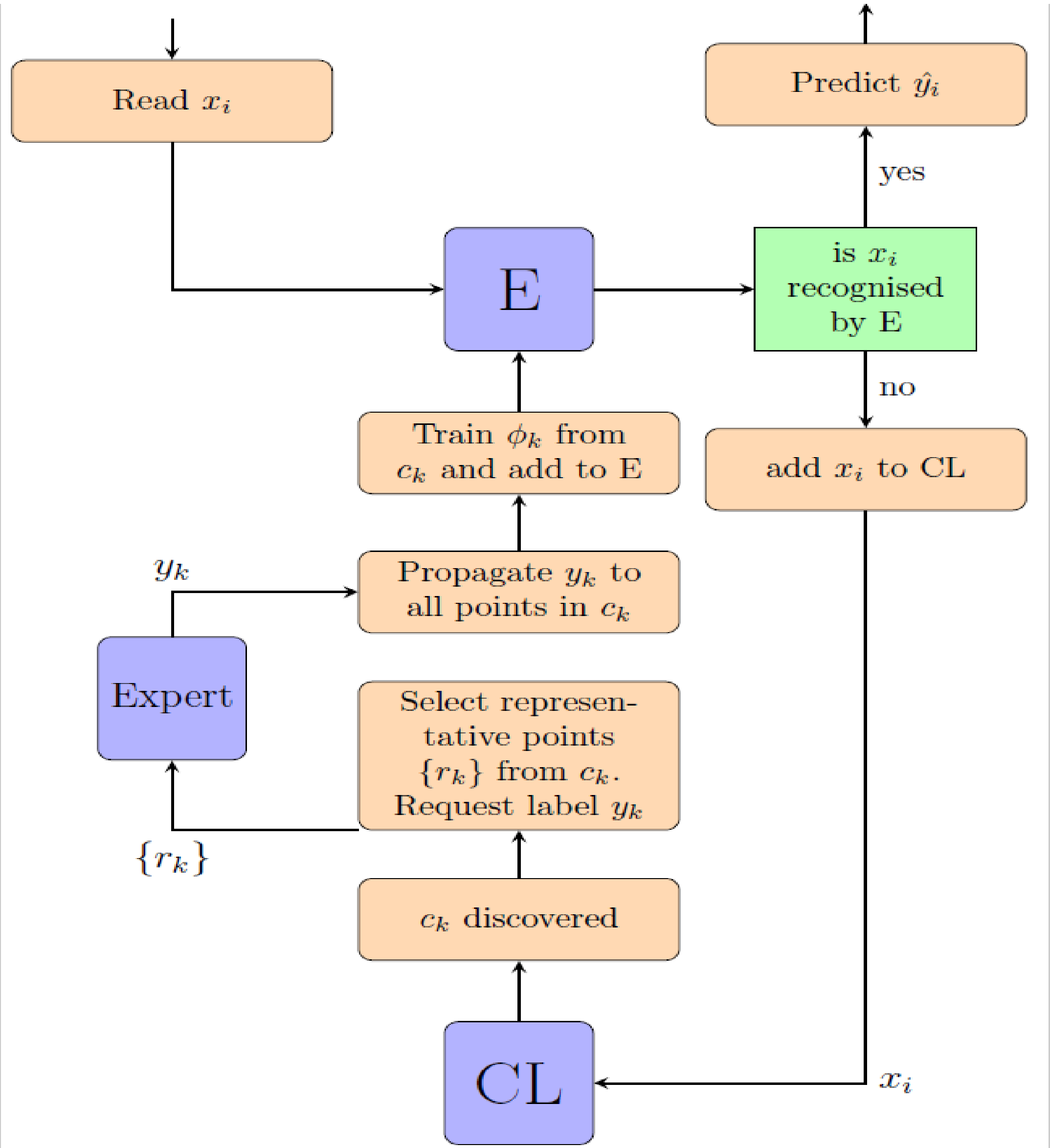
COCEL Framework

- Incoming point passed to Ensemble (E) of One Class Classifiers (OCCs)
- If point is recognised, prediction is made
- If point is not recognised, it is passed to stream clustering alg (CL)



COCEL Framework

- If a new cluster is discovered, representative samples passed to user for labelling
- New OCC trained on latest cluster and added to E
- Old OCCs which no longer make predictions are deleted from E



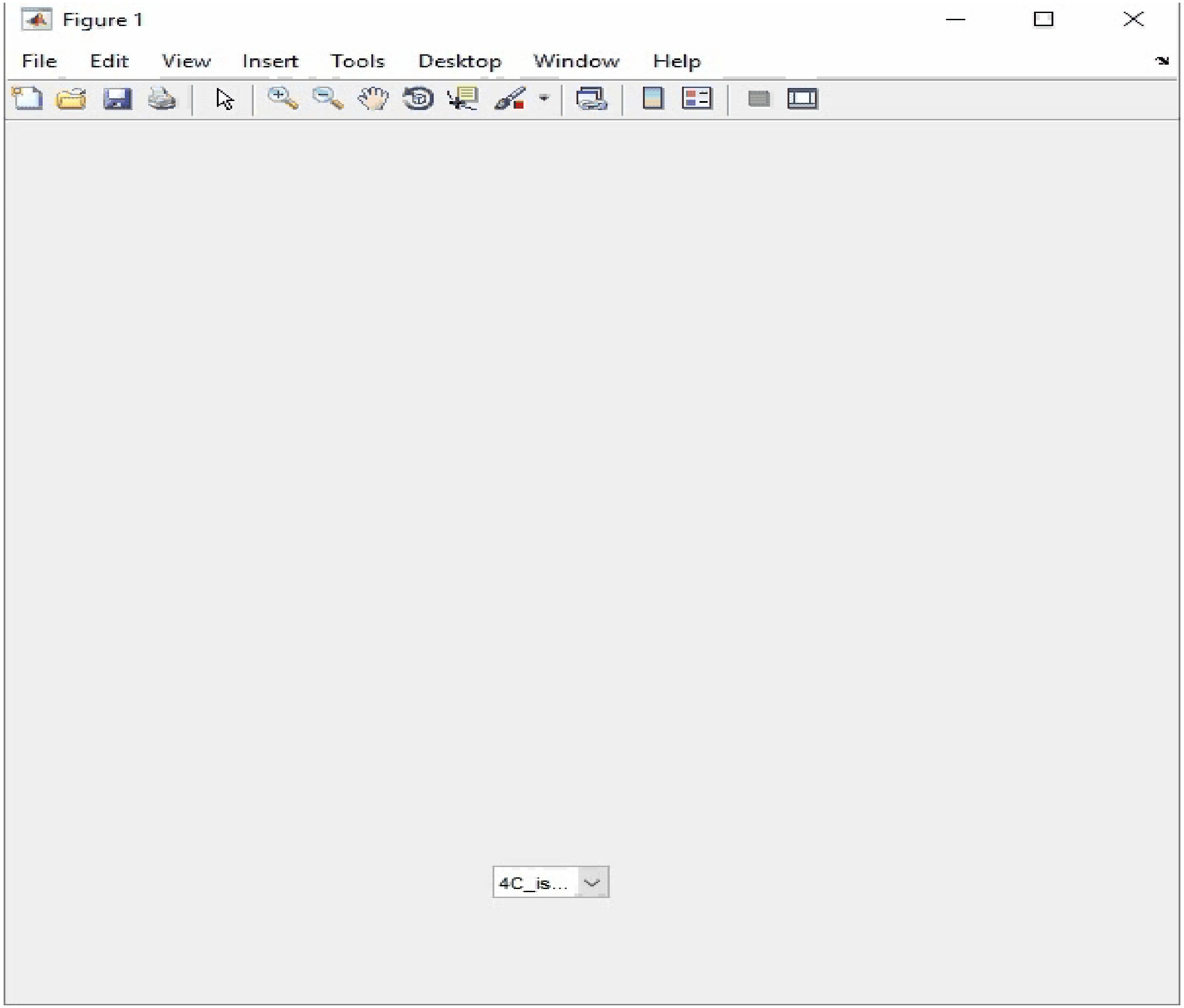
COCEL Experimental Study

- COCEL implementation:
  - Micro-classifiers as OCC (like micro-clusters but with an associated label)
  - MDSC as stream clustering algorithm
- COCEL compared with static ensemble
  - Static ensemble is trained but never updated as stream progresses

Demo: Synthetic Data

- Synthetic data stream, 4 classes in 2D
- 100K samples
- Simple but not trivial!
- Virtual Drift leading to Real Drift

Demo: Synthetic Data

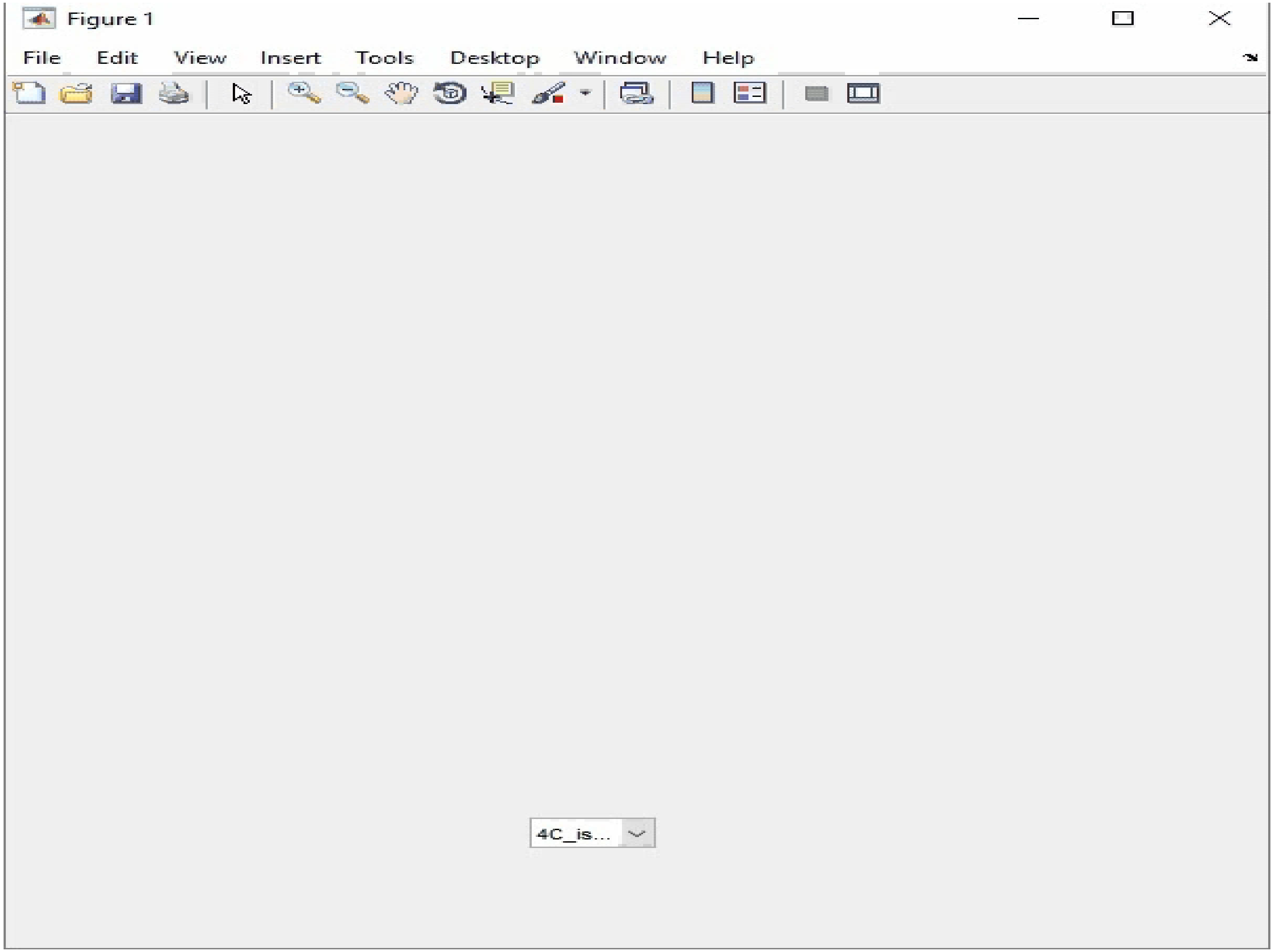


- ~96% accuracy; 560/100,000 labels

Demo: Network Intrusion

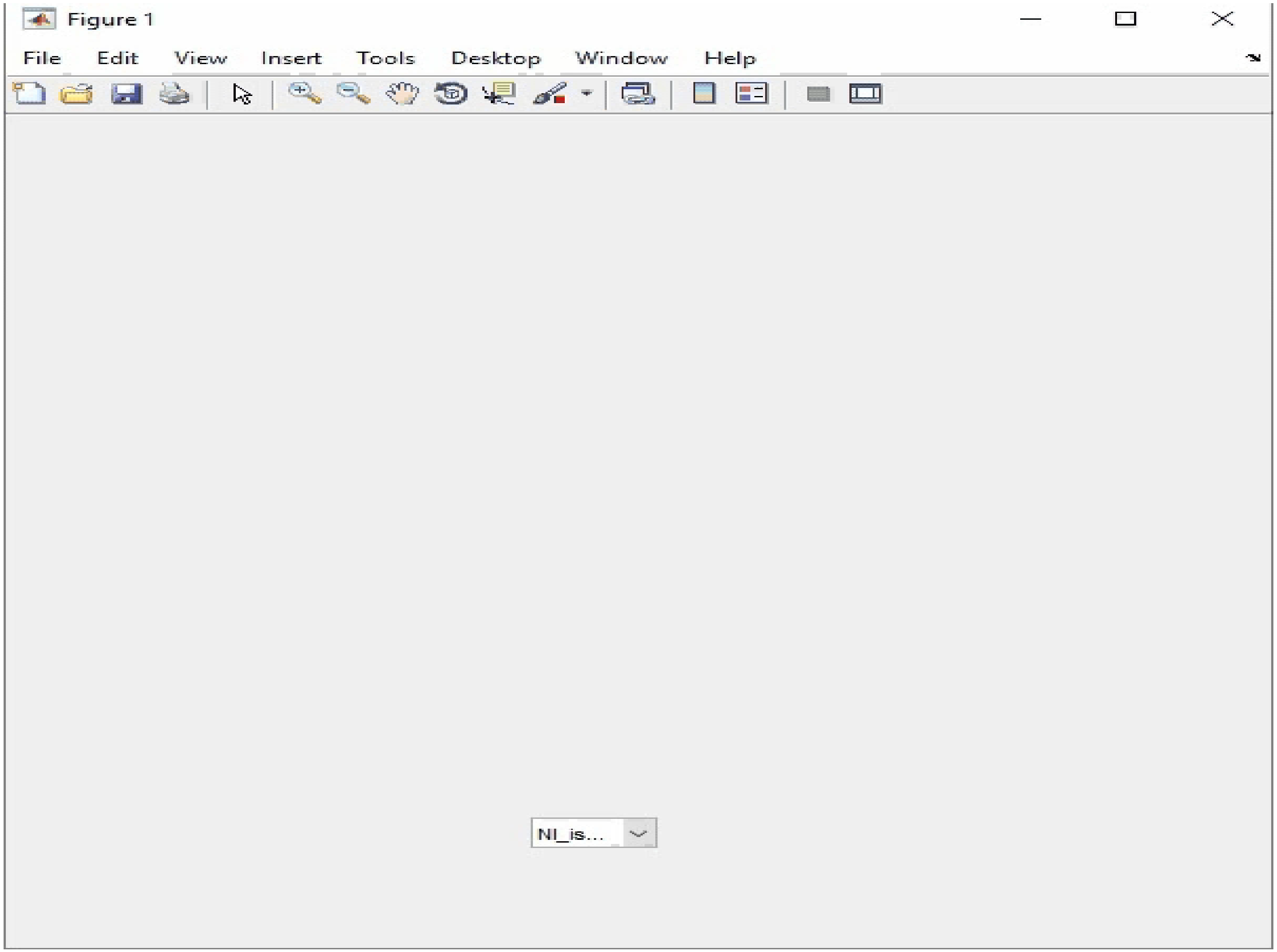
- Network Intrusion Data, 42 dimensions
- 1 “normal” class, 4 malicious classes
- Real Drift, Concept Evolution
- First 1,000 samples used as training set

Demo: Network Intrusion



- ~96% accuracy; 1102/200k labels (0.005%)

Demo: Network Intrusion No Training



- ~85% accuracy; 156/200k labels

Summary

- Data stream mining: interesting trend
- Stream clustering: Using ant colony behaviour
  - ACSC and MDSC
- Clustering and classification ensemble learning
  - COCEL