Image Classification Using an Ant Colony Optimization Approach

Tomas Piatrik and Ebroul Izquierdo

Multimedia & Vision Research Group
Queen Mary University of London

MULTIMEDIA & VISION RESEARCH LAB

Queen Mary University of London
Department of Electronic Enginnering

Overview

- Introduction
- Biologically Inspired Optimization Systems
  - Ant Colony Optimization (ACO)
- ACO - based image classifier
- Experimental Results
- Future work
  - AntTree - new model for clustering
- Conclusions

Image Classification

- **Data Mining tasks:** feature extraction,
  pattern recognition,
  prediction,
  classification,
  optimization,
  annotation, …
- **Image Classification**
  - the task is to assign images with same semantic content to predefined classes
  - two types of classification schemes: *supervised* and *unsupervised*.
- **Supervised classification**
  - requires relevance feed-back and/or correction from a human annotator
- **Unsupervised classification (clustering)**
  - does not require human intervention
- **The performance of the image classification algorithms relies on the efficient optimization techniques**

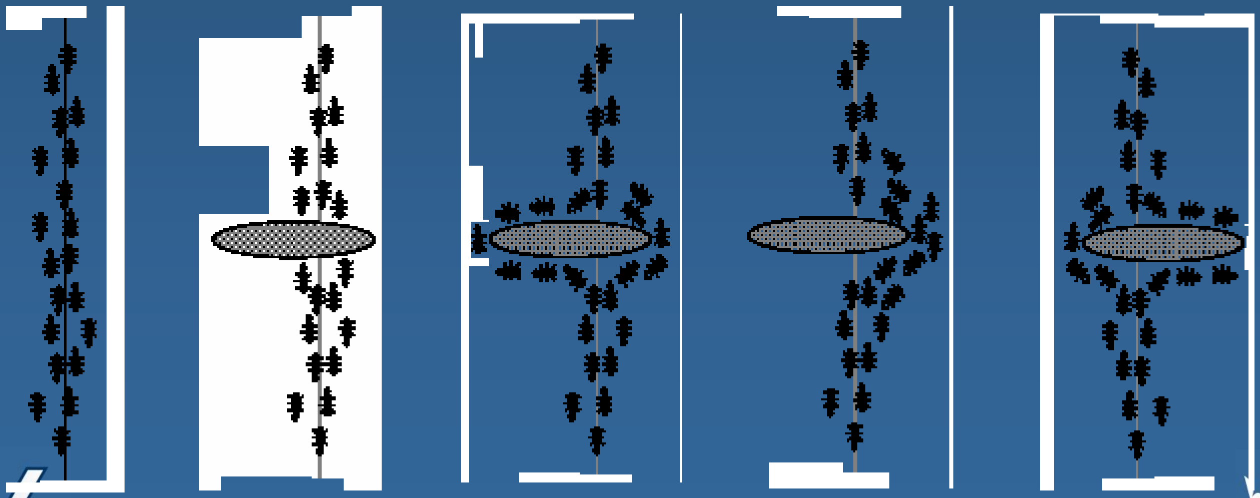
Biologically Inspired Optimization techniques

- **Optimization task**
  - process of adjusting the control variables to find the levels that achieve the best possible outcome.

Biologically Inspired systems: Artificial Immune Systems,
Particle Swarm,
Ant Colony Systems

Ant Colony Optimization (ACO)

- ACO is a meta-heuristic that uses strategies of real ants to solve optimization problems
- ACO is inspired by the observation of real ant colonies and is based on the indirect communication of a colony of simple agents, called (artificial) ants, mediated by (artificial) pheromone trails.
- An important and interesting behavior of ant colonies is their foraging behavior, and, in particular, how ants can find shortest paths between food sources and their nest.



Ant Colony System

- **The Ant System algorithm (AS) was first proposed to solving the Traveling Salesman Problem (TSP).**
  - Given a set of *n* cities and a set of distances between them, we call d_{ij} the length of the path between cities *i* and *j*.

o **The probability of choosing next *j* node:**

$$p_{ij}^{\,k}(t) = \begin{cases} \dfrac{(\tau_{ij}(t))^{\alpha}(\eta}{\sum\limits_{k \in allowed} (\tau_{ij}(t))^{\alpha}} \end{cases}$$

o **Heuristic information :**

$$\eta_{ij} = \frac{1}{d}$$

o **Pheromone value :**

$$\tau_{ij}(t) = \rho \cdot \tau_{ij}(t-1)$$

$$\Delta\tau_{ij} = \sum^{m} \Delta$$

$$\Delta\tau_{ij}^{\,k} = \begin{cases} \dfrac{Q}{L_k}, \text{if k - th ant use} \\ 0 \end{cases}$$

ACO-based Image Classifier

<u>The K-Means approach optimized by Ant Colony Optimization (ACO) System</u>

The procedure of K-Means algorithm:

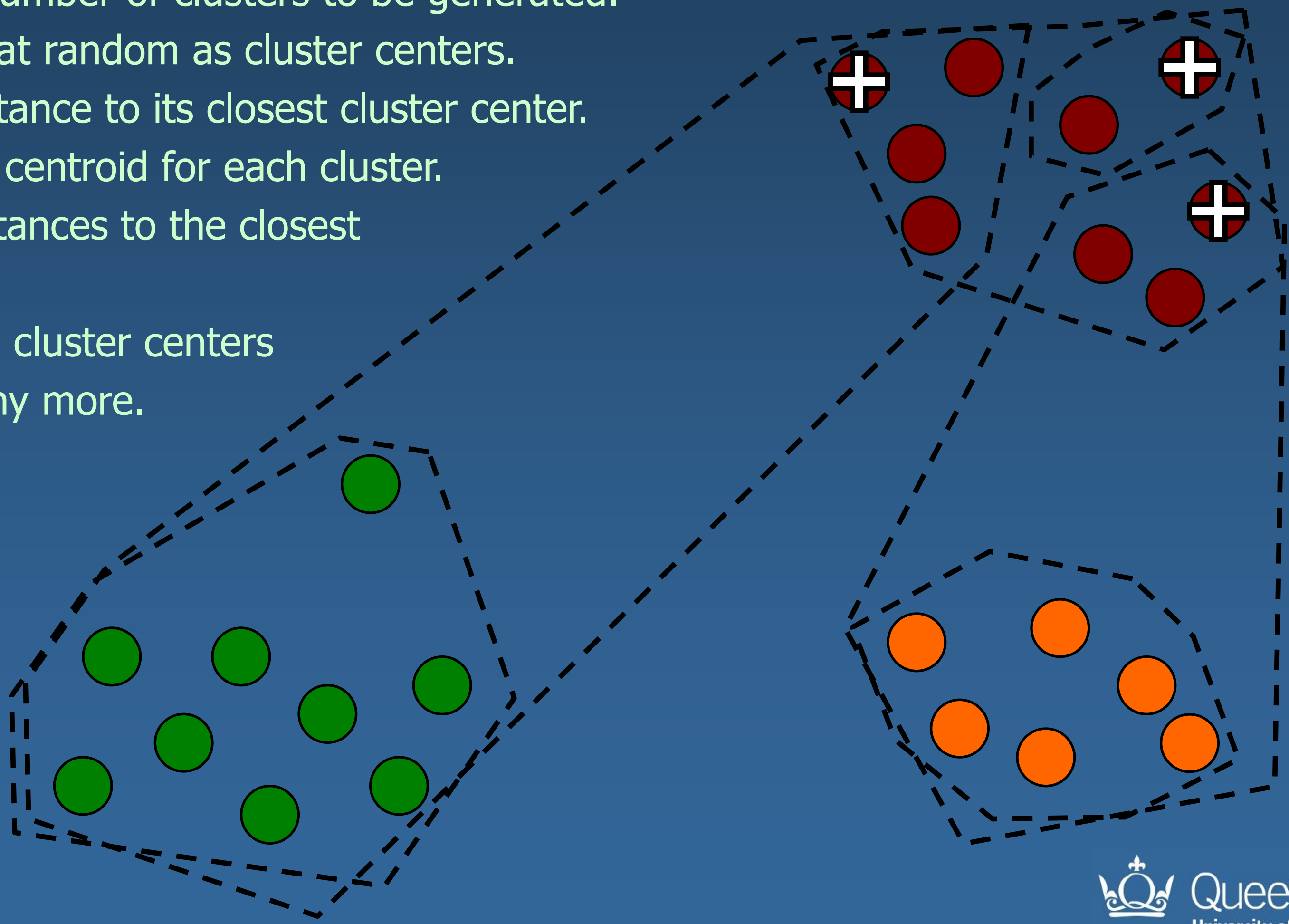
- Specify *k*, the number of clusters to be generated.
- Chose *k* points at random as cluster centers.
- Assign each instance to its closest cluster center.
- Recalculate the centroid for each cluster.
- Reassign all instances to the closest cluster center.
- Iterate until the cluster centers don't change any more.

ACO-based Image Classifier

The K-Means approach optimized by Ant Colony Optimization (ACO) System

The procedure of K-Means algorithm:

- Specify *k*, the number of clusters to be generated.
- Chose *k* points at random as cluster centers.
- Assign each instance to its closest cluster center.
- Recalculate the centroid for each cluster.
- Reassign all instances to the closest cluster center.
- Iterate until the cluster centers don't change any more.



ACO-based Image Classifier

- ACO plays its part in assigning each image to a cluster and each ant is giving its own classification solution.

- ***Step 1***: Initialize pheromone level to 1, heuristic information, the number of clusters to *K* and number of ants to *m*.
- ***Step 2***: For each ant, let each image *x* belong to one cluster with the probability.
- ***Step 3:*** Calculate new cluster center; If the new cluster centers converge to the old ones, go to next step otherwise, go to Step 2.

$$\eta_{(X_i,C_j)} = \frac{B}{Dist(X}$$

$$p_{(X_i,C_j)} = \frac{\tau^{\alpha}_{(X_i,C_j)}\eta^{\beta}}{\sum^{K}\tau^{\alpha}_{(X_i,C_j)}\eta}$$

- ***Step 4:*** Update the pheromone level on all images according to the quality of the solution.

$$\tau_{(X_i,C_j)}(t) = \rho\cdot\tau_{(X_i,C_j)}(t-1) + \sum^{m}$$

$$\Delta\tau_{(X_i,C_j)} = \begin{cases} \dfrac{N\cdot\sum_{j\neq i} Dist(C_i,C_j)}{\sum_{i=0}^{K} sumDist(C_i)}, \text{if } X_i \end{cases}$$

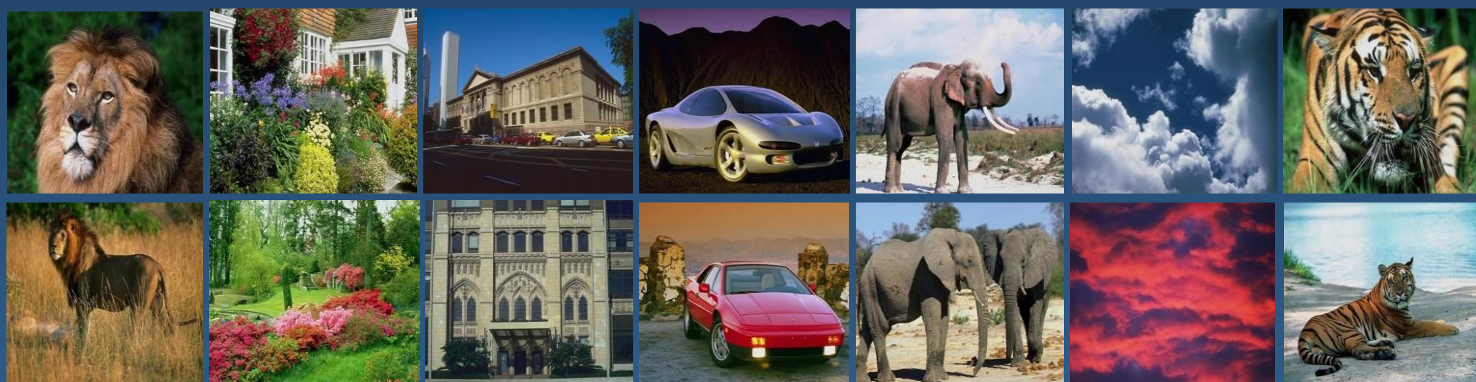
- ***Step 5:*** If the termination criterion is satisfied go to next step otherwise, go to Step 2.
- ***Step 9***: Output the optimal solution.

Experimental results

- For feature representing of images was used MPEG7 - Color Layout Descriptor (CLD)

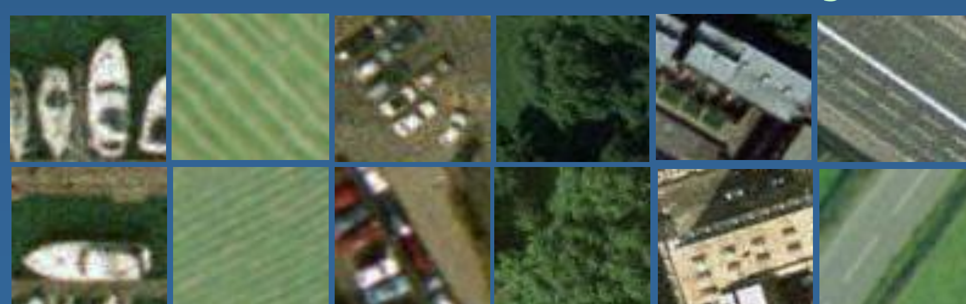
$$Sim(X_i,X_j)=\sqrt{\sum_{k=1}^{28}(X_i(k)-X_j(k))^2}+\sqrt{\sum_{k=29}^{43}(X_i(k)-X_j(k))^2}+\sqrt{\sum_{k=44}^{58}(X_i(k)-X_j(k))^2}$$

- **The Corel image database** - 700 images with 7 semantic concepts



Lions Rural Buildings Cars Elephants Clouds Tigers

- **The Window on the UK 2000" database -** 390 images with 6 sets



Boats Fields Vehicles Trees Buildings Roads

Experimental results

The Corel image database

Class1	Class2	K-Means			ACO-Classifier		
		Precision [%]	Recall [%]	Accuracy [%]	Precision [%]	Recall [%]	Accuracy [%]
Rural	Buildings	69	**93**	76	**72**	**93**	**78.5**
Lions	Cars	60	**52**	62	**68**	51	**72**
Elephants	Clouds	82	93	86	**84**	**95**	**88.5**
Lions	Rural	57	**68**	59	**65**	**68**	**66**
Buildings	Cars	70	**94**	76.5	**74**	90	**79.5**
Tigers	Clouds	**79**	73	82.5	76	**76**	**84**

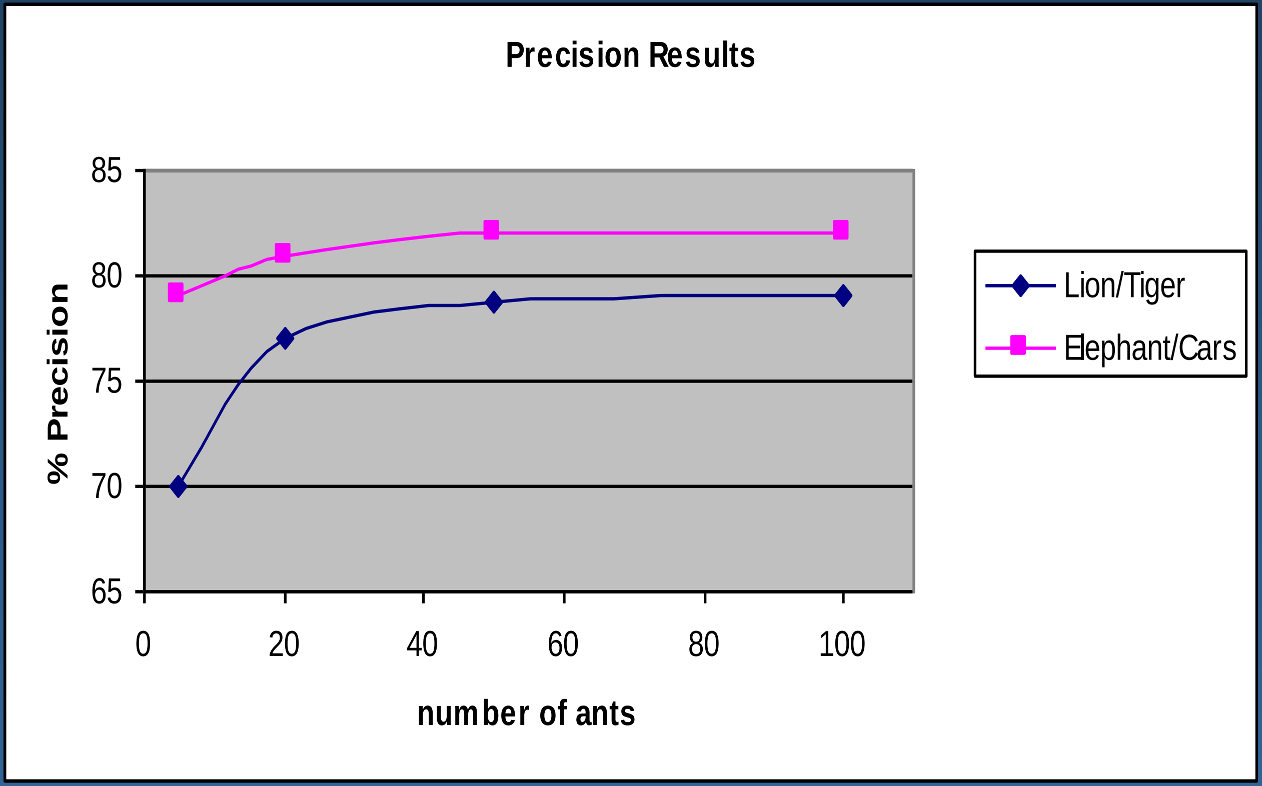
Experimental results

The Window on the UK 2000” database

Class1	Class2	K-Means			ACO-Classifier		
		Precision [%]	Recall [%]	Accuracy [%]	Precision [%]	Recall [%]	Accuracy [%]
Boat	Vehicle	68	65	62	**73**	**70**	**70**
Tree	Building	76	**67**	85	**98**	66	**90**
Building	Road	60	**45**	60	**78**	44	**71**
Field	Vehicle	87	**68**	93	**90**	**68**	**94**
Field	Tree	98	60	96	**100**	**61**	**97**
Road	Vehicle	68	52	73	**78**	**57**	**79**
Tree	Road	88	**54**	82	**96**	**54**	**90**

Experimental results

Classification results on visual similar / dissimilar images



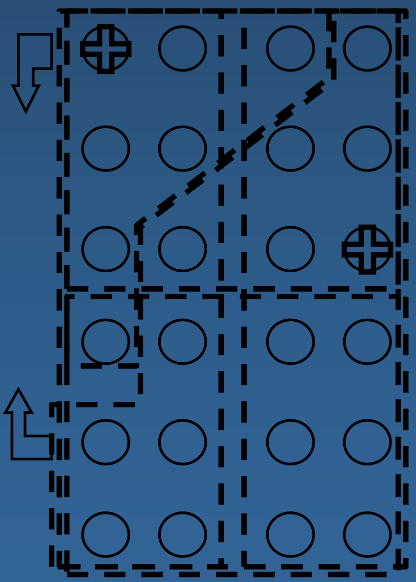
Optimization of COP-K-Means

COP-K-Means

- semi-supervised variant of K-Means, where initial background knowledge, provided in the form of constraints between instances in the dataset, is used in the clustering process.

- Two types of constraints: *must-link*
  cannot-link

- Goal - minimization of an objective function:



Evaluation of COP-K-Means

- Advantages :
  - less computational overhead
  - high efficiency for large data sets

- Weakness :
  - high dependency of random initialization of cluster centers - often terminates at a *local optimum*

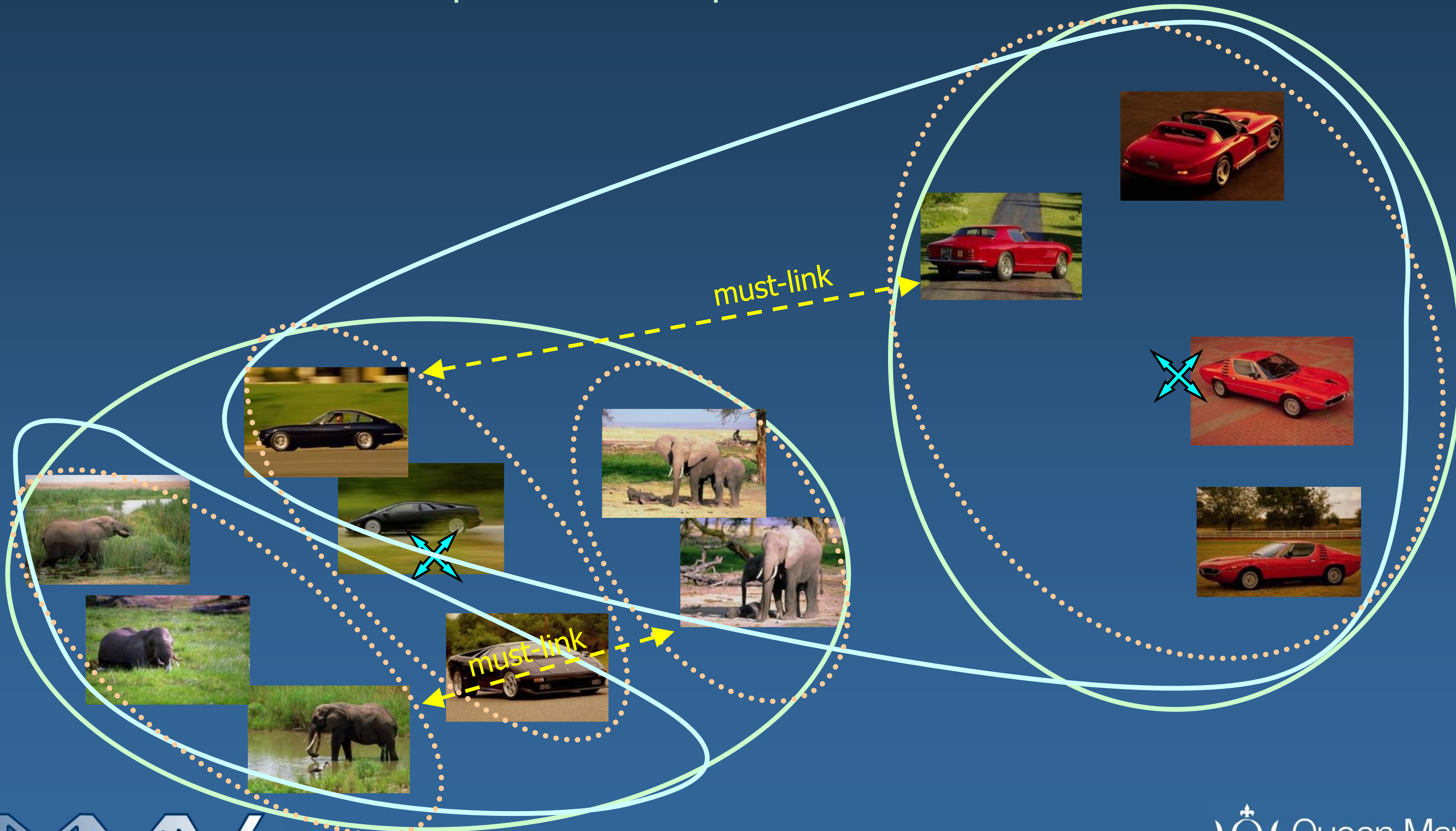
    Ant Colony Optimization

  - sensitivity to outliers and noise
  - constrains in feature space don't interpret semantic information

    New constraints representation in Feature space

Future Work

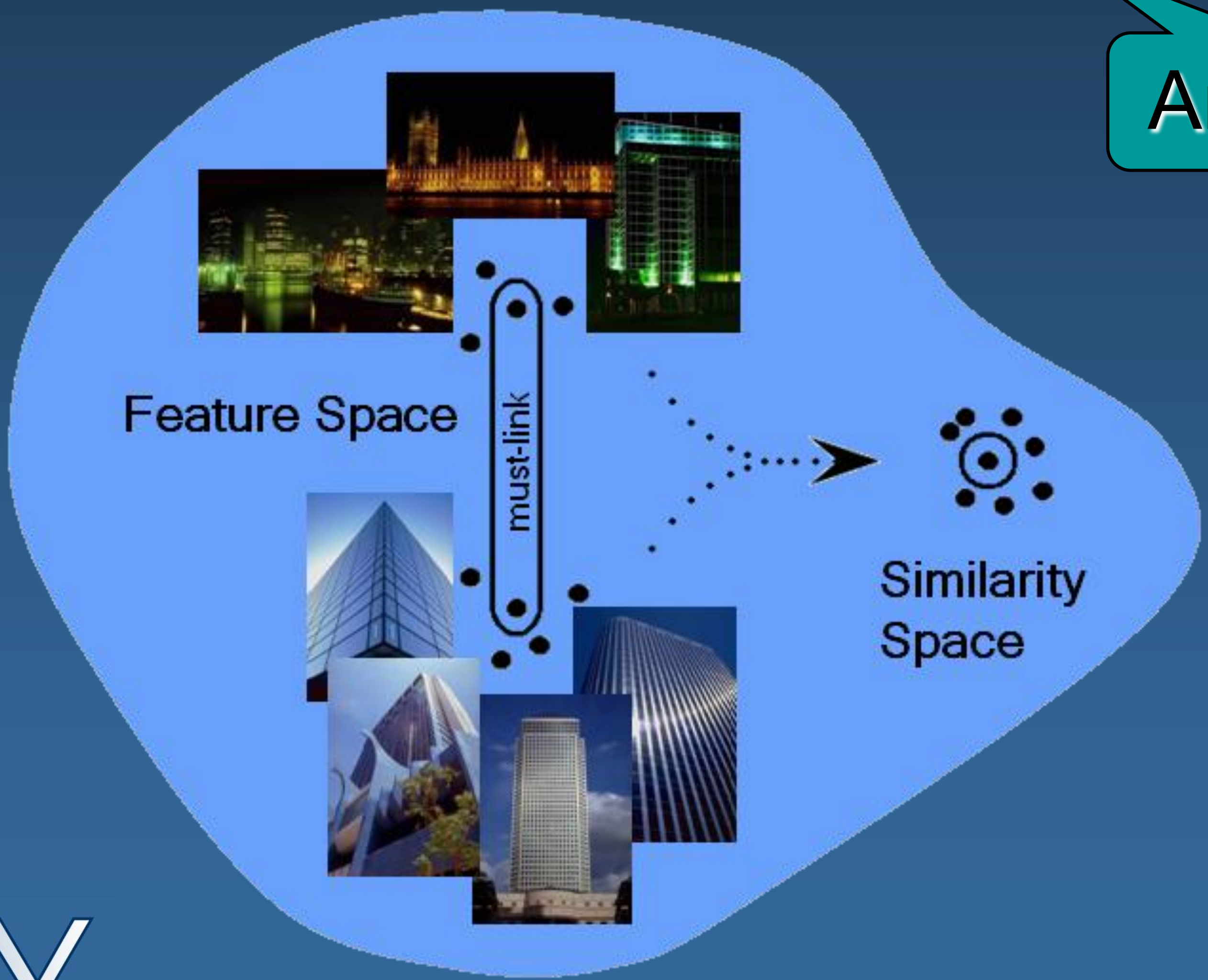
- sensitivity to outliers and noise
- constrains in feature space don’t interpret semantic information



Future Work

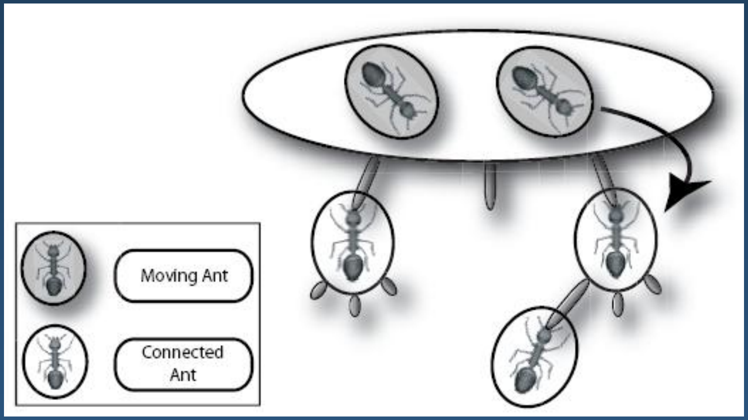
- sensitivity to outliers and noise
- constrains in feature space don’t interpret semantic information

AntTree



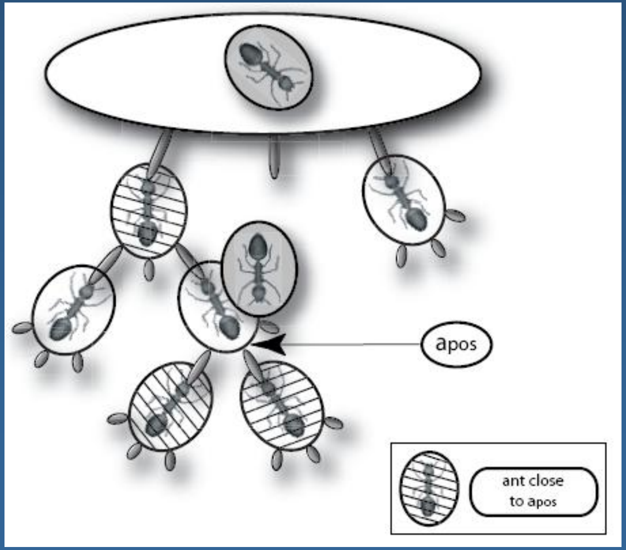
AntTree: New model for clustering

- General principles



each ant: represents node of tree (data)

- outgoing link; a_i can maintain toward another ant
- incoming links; other ants maintain toward a_i
- a_o support, a_{pos} *position of moving ant*



- Main algorithm

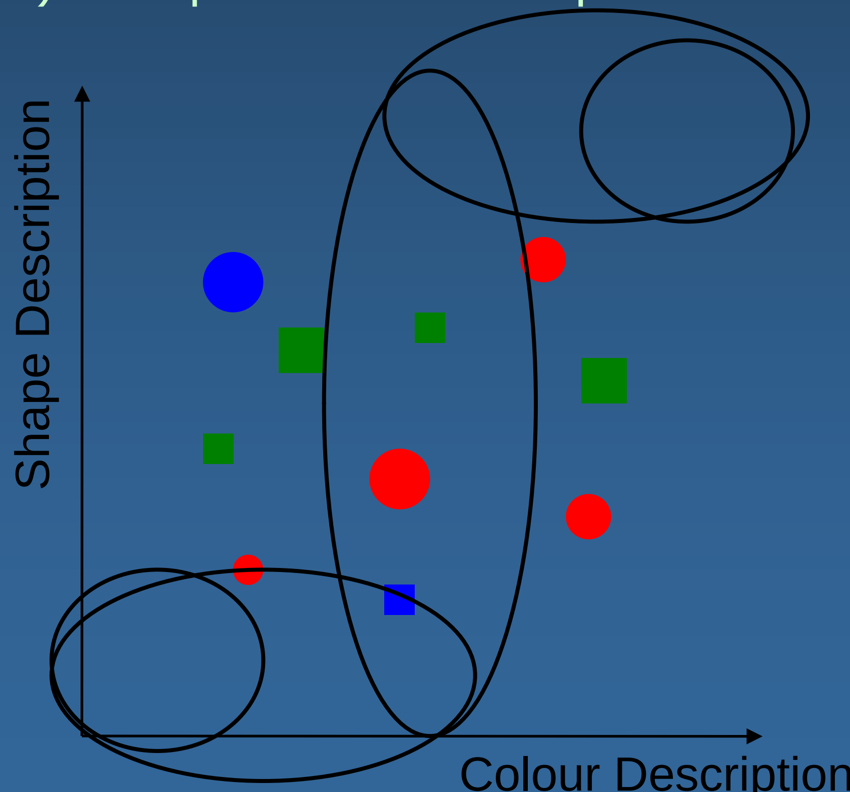
1. all ants placed on the support;
   initialization: $T_{sim}(a_i)=1$, $T_{dissim}(a_i)=0$
2. While there exists non connected ant a_i Do
3. If ai is located on the support Then ***Support case***
4. Else ***Ant case***
5. End While

Conclusion and Future work

- The ACO makes the K-Means algorithm less dependent on the initial parameters; hence it makes it more stable and efficient

Next steps:

- Integration of AntTree to training process of ACO/COP-K-Means
- Implementation of Ant Colony Optimization with Multi-descriptor space (MPEG-7 descriptors) and optimization of the performance.



Conclusion and Future work

- The ACO makes the K-Means algorithm less dependent on the initial parameters; hence it makes it more stable and efficient

Next steps:

- Integration of AntTree to training process of ACO/COP-K-Means
- Implementation of Ant Colony Optimization with Multi-descriptor space (MPEG-7 descriptors) and optimization of the performance.
    - *Feature Subset Selection Using Ant Colony Optimization*

Thank you for your attention!

tomas.piatrik@elec.qmul.ac.uk