Clustering Ensemble

Ke Chen

Department of Computer Science, The University of Manchester

Ke.Chen@manchester.ac.uk

Outline

Introduction
Motivation and strength

Problem Formulation and Generic Solution
Formal description, main criteria, generic solution, partition generation, existing algorithms

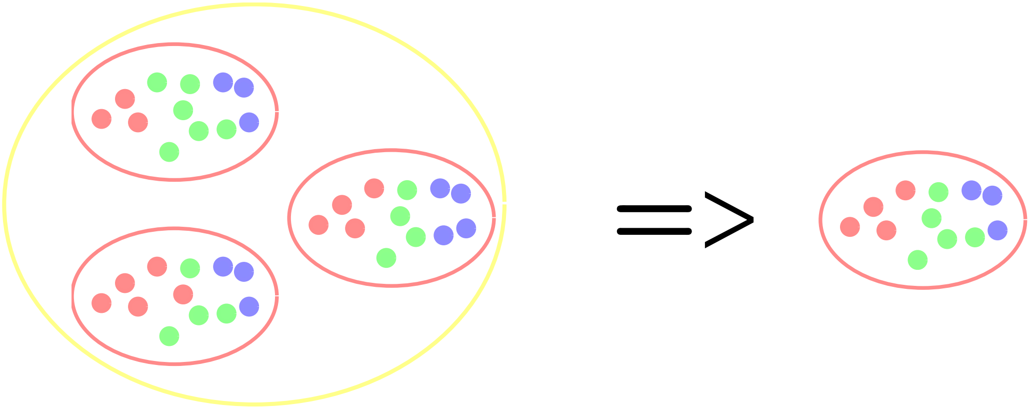
Evidence-accumulated Clustering Ensemble
Main steps, algorithmic description and illustrative example

Weighted Clustering Ensemble
Internal versus external validity indexes, algorithm and illustrative example/application

- **Motivation**
  - Combination of multiple learners in supervised learning takes advantage of complement and diversity among multiple learners, which has turned out to be one of the most successful learning strategy.
  - For clustering analysis, clustering ensemble bears the same motivation to form an effective unsupervised learning strategy.
  - The idea behind clustering ensemble is to find out a robust consensus partition from multiple ones achieved on different conditions or with different methods by utilising complementary perspectives of data.
- **Strength**
  - Knowledge reuse: consensus partition can be obtained from previous partitions on partition level regardless of how and when to be generated.
  - Distributed computing: individual partitions can be obtained independently.
  - Privacy: only cluster information in individual partitions is required for consensus rather than features or attributes of data points.

- **Clustering ensemble problem**: Given a dataset X and a set of n partitions on X, $\mathbb{P} = \left\{P^{(1)}, P^{(2)}, \cdots, P^{(n)}\right\}$ where $P^{(i)} = \left\{C_1^{(i)}, C_2^{(i)}, \cdots, C_{k_i}^{(i)}\right\}, i = 1, 2, \cdots, n$, find out a new consensus partition, P^*, that uses the information of all n partitions in $\mathbb{P}$.
- **Criteria for consensus partition**
  - Robustness: the consensus partition should have better performance than each of individual partitions to be combined in some sense.
  - Consistency: the consensus partition should be similar to the individual partitions overall and preserve their commonality.
  - Stability: the consensus partition should be less sensitive to outliers and noise than each of individual partitions to be combined.
  - Novelty: the consensus partition should be a different partition that cannot be obtained by any clustering algorithms used to generate those individual partitions for combination.

- **Generic solution to clustering ensemble**
  1. Generate different individual partitions to be combined
  2. Combine those partitions to form a consensus partition



Partition Generation

- Different representations: use different feature sets or multi-views of raw data
- Different clustering algorithms: use different clustering algorithms that have different biases in clustering analysis
- Different similarity/distance metrics: use different similarity/distance metrics that might approximate the "true" metric in clustering analysis
- Different hyper-parameters and initialisation: use different hyper-parameter values and initialisation conditions, e.g., different number of clusters and initial cluster centres in K-means
- Subspace projection: use different dimension reduction techniques
- Bootstrap: use random subsets of raw data via re-sampling

Clustering Ensemble Algorithm

- **Co-occurrence based clustering ensemble**: use the clusters obtained from each individual partition and the coincidence of those clusters in different partitions to find out a consensus partition.
  - Cluster alignment and majority-voting on partitions
  - Evidence accumulation via co-association matrix
  - Graph and hyper-graph partitioning
  - Maximum mutual information among partitions
  - Mixture of multinomial distribution on partitions
- **Median partition clustering ensemble**: for a set of partitions and a given similarity metric, find the "median" partition that maximises the overall similarity between this optimal partition and any one in the partition set.

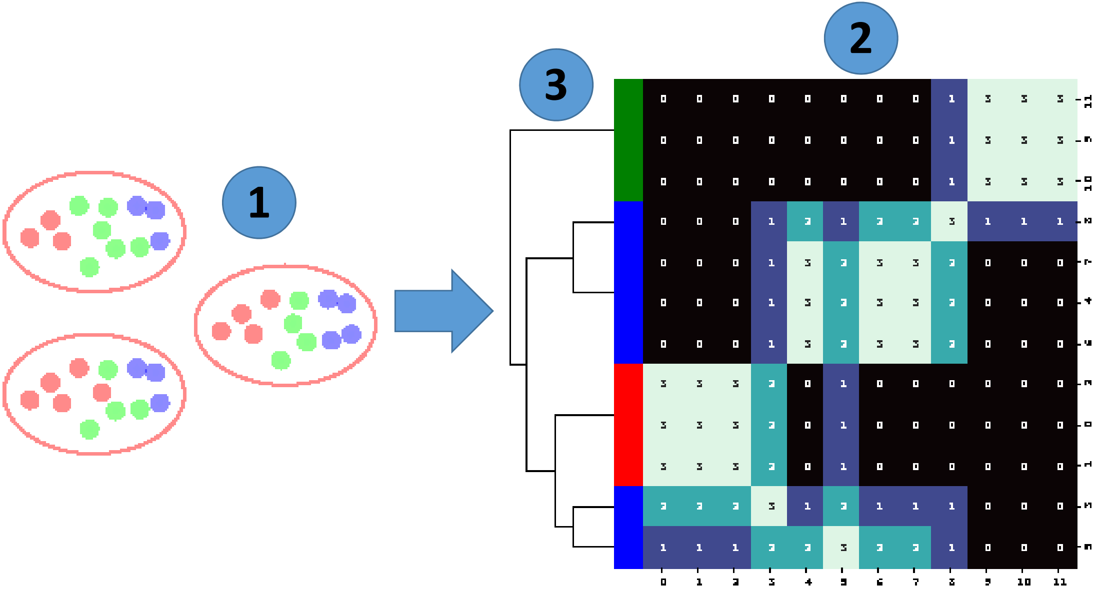
Key Idea of Evidence-accumulated Clustering Ensemble

1. **Different partition creation**: Given a dataset X with $|X|$ data points, use a proper manner to create a set of n partitions on X, $\mathbb{P} = \left\{P^{(1)}, P^{(2)}, \cdots, P^{(n)}\right\}$.
2. **Evidence accumulation with co-association matrix**: data points belonging to a "natural" cluster very likely to be co-located in the same cluster in different partitions. Such information can be encoded in the co-association matrix on any pair of points:

$$\mathcal{C}_{|X|\times|X|} = \{c(q,r)\}\,, \quad q = 1, 2, \cdots, |X|, \;\; r = 1, 2, \cdots, |X|$$

   where $c(q,r)$ is counted by the number of times a pair of points $(\boldsymbol{x}_q, \boldsymbol{x}_r)$ assigned to the same cluster among n different partitions in $\mathbb{P}$, and $0 \leq c(q,r) \leq n$.
3. **Finding out the optimal consensus partition**: Treat the co-association matrix a collective "similarity" matrix and convert it into a "distance" matrix. Then apply Agglomerative algorithm to this distance matrix to generate the consensus partition.

Evidence-accumulated Clustering Ensemble



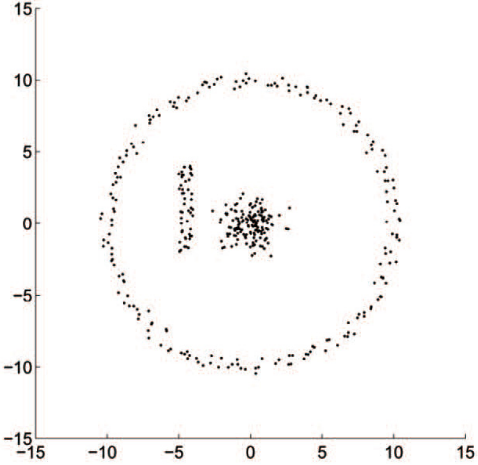
Evidence-accumulated Clustering Ensemble

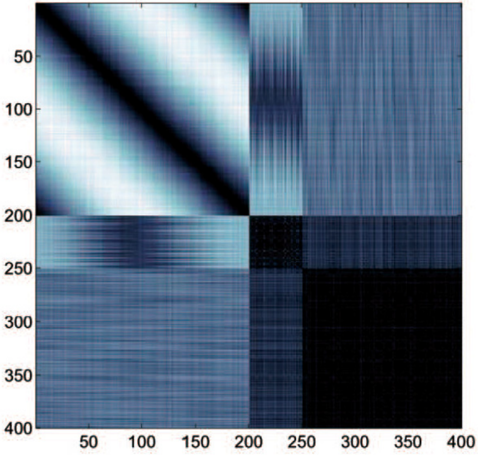
Input: Data set X with $|X|$ data points

- **Initialisation**
  - With a proper manner, create a set of n different partitions on X, $\mathbb{P} = \left\{P^{(1)}, P^{(2)}, \cdots, P^{(n)}\right\}$, where $P^{(i)} = \left\{C_1^{(i)}, C_2^{(i)}, \cdots, C_{k_i}^{(i)}\right\}$, $i = 1, 2, \cdots, n$.
  - Set the co-association matrix $\mathcal{C} = \{c(q, r)\}_{|X|\times|X|}$ to a `null` matrix; i.e., $c(q, r) = 0$, $q = 1, 2, \cdots, |X|$, $r = 1, 2, \cdots, |X|$.
- **Evidence Accumulation**
  - For each partition, $P^{(i)} \in \mathbb{P}$ $(i = 1, 2, \cdots, n)$, update the co-association matrix $\mathcal{C} = \{c(q, r)\}_{|X|\times|X|}$: $c(q, r) \leftarrow c(q, r) + \frac{1}{n}$, for $q, r = 1, 2, \cdots, |X|$ if the pair of data points $(\boldsymbol{x}_q, \boldsymbol{x}_r)$ belongs to the same cluster in $P^{(i)}$.
- **Optimal Consensus Partition Generation**:
  - Choose one proper cluster-distance measure from d_{SL}, d_{CL} or d_{GA}. Convert co-associate matrix to distance matrix $\mathcal{D} = \{d(q, r)\}_{|X|\times|X|}$: $d(q, r) = 1 - c(q, r)$. Apply Agglomerative algorithm to $\mathcal{D}$ and find out the longest K-cluster lifetime from its dendrogram tree. Output the corresponding partition, P^*.

Illustrative Example: 3-Cluster Dataset

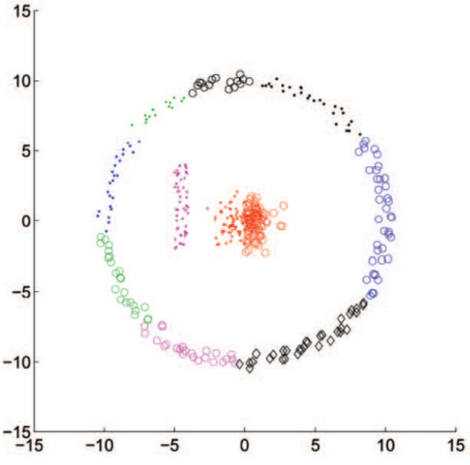
- 400 data points: outer ring (200 points), rectangular shaped cluster (50 points), and 2D Gaussian cluster (150 points). Similarity matrix achieved with Euclidean distance.

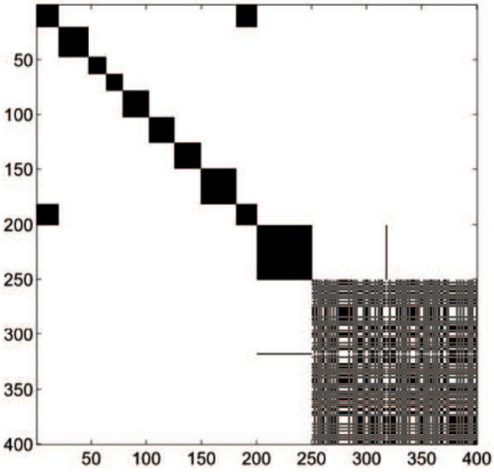




Illustrative Example: 3-Cluster Dataset

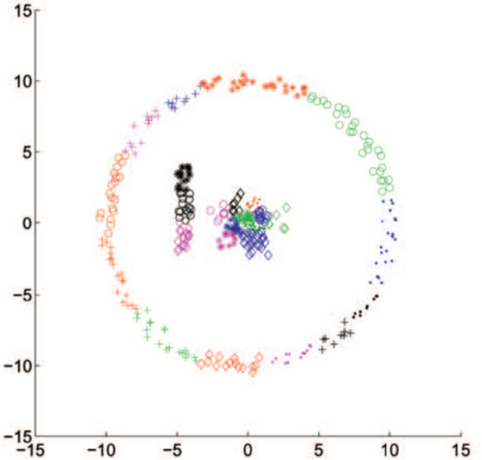
- Apply K-means ($K = 11$). Generate the co-association matrix based on only this single partition of 11 clusters.

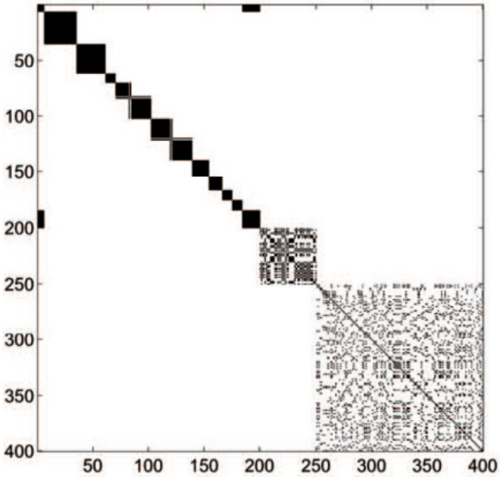




Illustrative Example: 3-Cluster Dataset

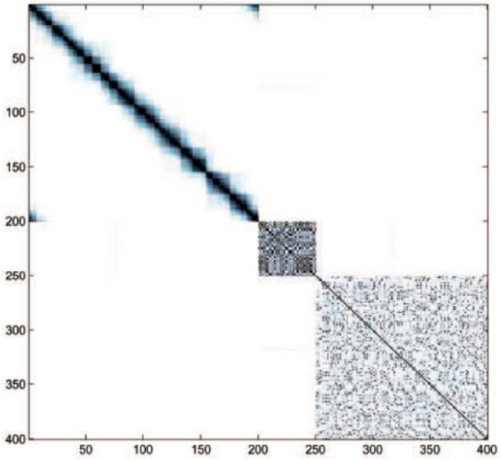
- Apply K-means ($K = 25$). Generate the co-association matrix based on only this single partition of 25 clusters.

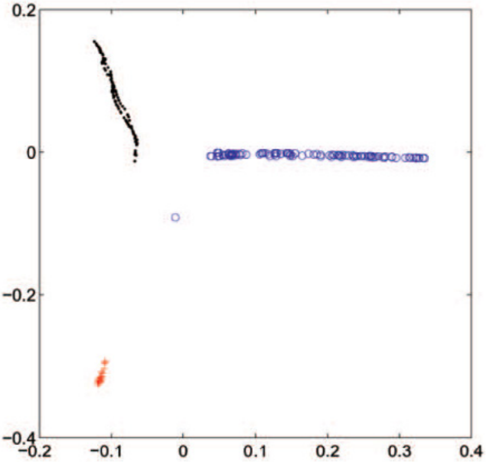




Illustrative Example: 3-Cluster Dataset

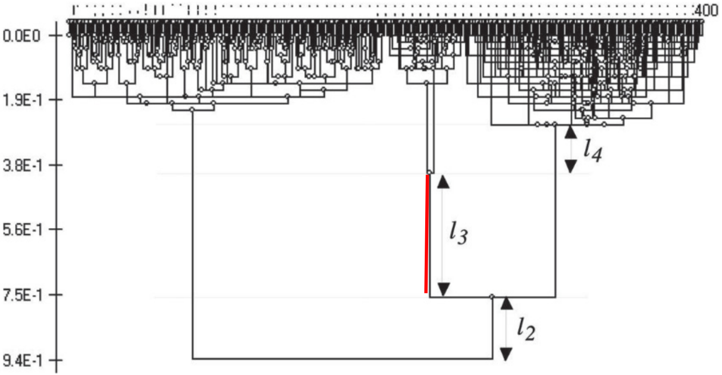
- 30 partitions created with random initialisation and K randomly chosen in the interval $[10, 30]$; co-association matrix based on all 30 partitions and 2-D MDS embedding.

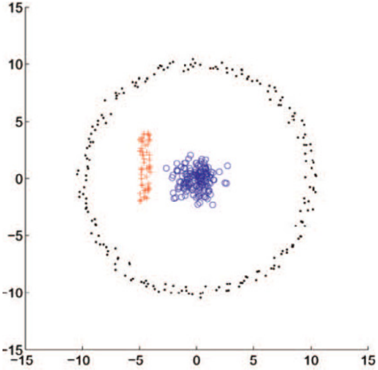




Illustrative Example: 3-Cluster Dataset

- Apply Agglomerative algorithm to the collective distance matrix to generate the dendrogram tree. Find the optimal partition of 3 clusters (longest K-cluster lifetime).





- **Motivation**
  - Evidence-accumulated method treats all the partitions equally but different partitions have unequal amount of useful information.
  - Cluster validity indexes can evaluate the quality of partitions from different perspectives.
  - Therefore, such indexes may be used to highlight the partitions of "useful" information and diminish the partitions of "useless" information in clustering ensemble.
  - By choosing the proper cluster validity indexes, their values are used to weight co-association matrix for a better consensus partition.

- **Internal Cluster Validity Index**
  - Internal validity index is an evaluation function, $g(P)$, working on a single partition, P, to measure the quality with "common sense" or "prior knowledge".
  - Typical internal indexes: scatter-based F-ratio, rate-distortion, Davies-Bouldin index (DBI), Bayesian information criterion (BIC), silhouette Coefficient, minimum description length (MDL), stochastic complexity (SC) and modified Huber's Γ index (MHΓ)
- **External Cluster Validity Index**
  - External validity index is an evaluation function, $g(P, \bar{P})$, that measures the quality by comparing a single partition, P, against a "ground-truth" or a reference partition, $\bar{P}$, that is not used in obtaining P in clustering analysis.
  - Typical external indexes: Rand index, adjusted Rand index, pair counting index, information theoretic index, set matching index, DVI index and normalised mutual information index (NMI)

Fact: $g(P)$ and $g(P, \bar{P})$ are normalised within $[0, 1]$; the larger its value the higher quality a partition.

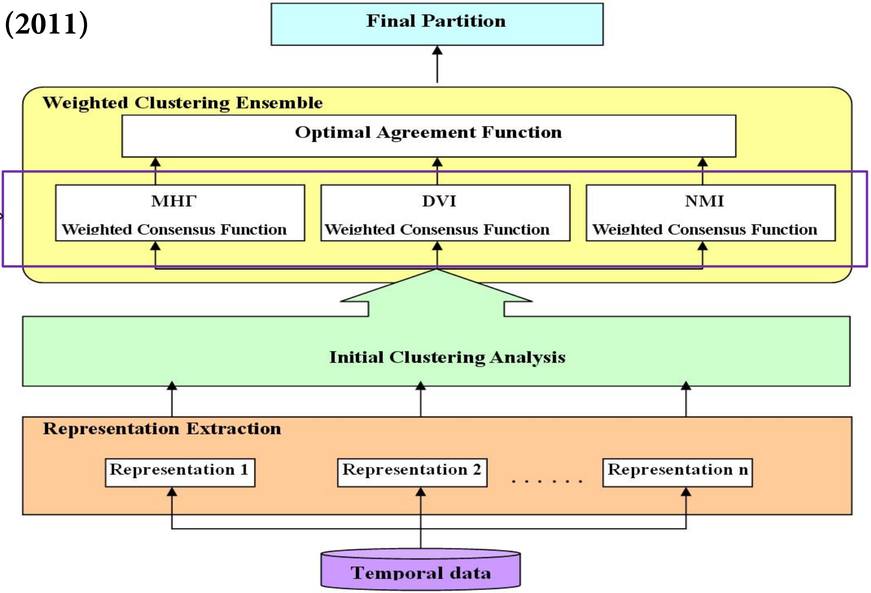
- **Weighted Co-association Matrix**
  - The only difference between evidence-accumulated and weighted clustering ensemble lies in collective "similarity" matrix.
  - When an internal validity index is used, the weight for any $P^{(i)} \in \mathbb{P}$ is generated by $w^{(i)} = g\left(P^{(i)}\right)$.
  - When an external validity index is used, the weight for any $P^{(i)} \in \mathbb{P}$ is generated by $w^{(i)} = \frac{1}{n-1}\sum_{j=1, j\neq i}^{n} g\left(P^{(i)}, P^{(j)}\right)$.
  - By choosing m cluster validity indexes to generate m weights, $w_1^{(i)}, w_2^{(i)}, \cdots, w_m^{(i)}$, for each $P^{(i)} \in \mathbb{P}$, the weighted co-association matrix, $\hat{\mathcal{C}}$, initially set to `null` is updated for $i = 1, 2, \cdots, n$:

$$\hat{\mathcal{C}} = \{\hat{c}(q,r)\}_{|X|\times|X|}: \;\; \hat{c}(q,r) \leftarrow \hat{c}(q,r) + \frac{1}{n}\left(\frac{\sum_{k=1}^{m} w_k^{(i)}}{m}\right)$$

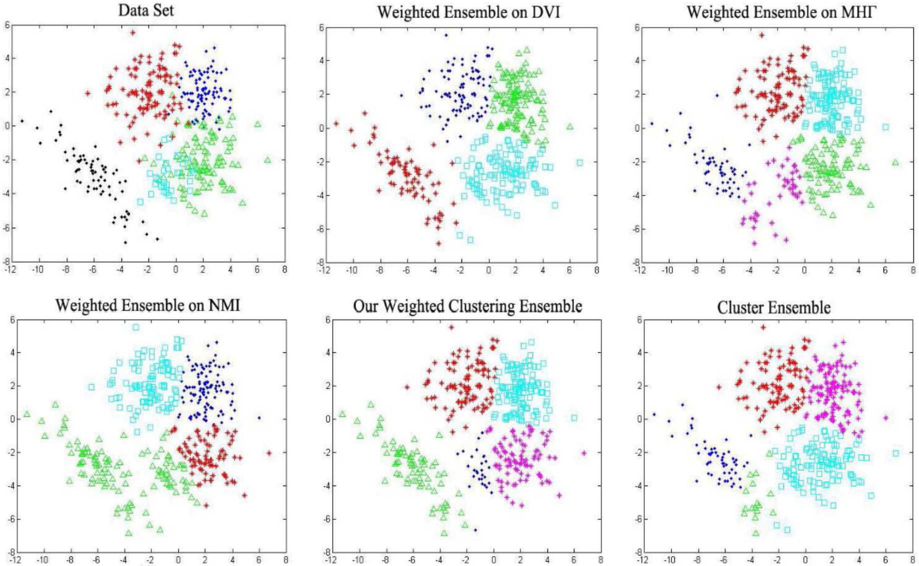
Weighted Clustering Ensemble

Yang & Chen (2011)

Use validity index values as weights

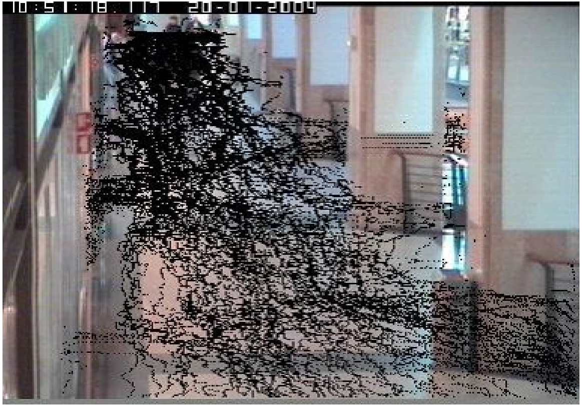


Illustrative Example: 5-Cluster Dataset (20 K-means partitions + d_{SL})



Application: Trajectory Clustering for Knowledge Discovery

- CAVIAR Database: annotated video clips of pedestrians, 222 moving trajectories
- Temporal Representation: PCF, DCF, PLS and PDWT
- Partition Creation: K-means with random initialisation and K randomly chosen from the interval $[4, 20]$. Totally, 320 partitions (80 paritition/representation)



Application: Trajectory Clustering for Knowledge Discovery

- Optimal Partition: Group-average cluster distance leading to P^* of 16 clusters

Reference

If you want to deepen your understanding and learn something beyond this lecture, you can self-study the optional references below.

[Fred & Jain, 2005] Fred A. and Jain A.K. (2005): Combining multiple clusterings using evidence accumulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 6, pp. 835-850.

[Yang & Chen, 2011] Yang Y. and Chen K. (2011): Temporal data clustering via weighted clustering ensemble with different representations. *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 2, pp. 307-320.

[Wang et al., 2009] Wang K., Wang B. and Peng L. (2009): CVAP: Validation for cluster analyses. *Data Science Journal*, Vol. 8, pp. 88-93.