Multimodality in Data Clustering: Application to Video Summarization

Aristidis Likas
CSE Department
University of Ioannina
arly@cs.uoi.gr

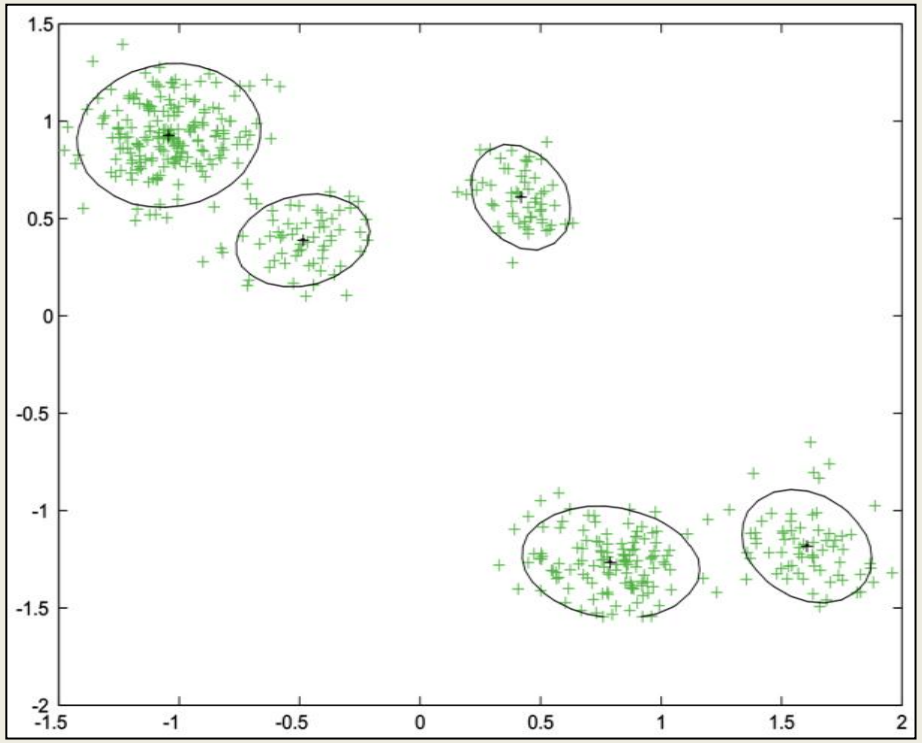
AIAI 2016 Conference

Outline

- Clustering multimodal data
- Multimodality as a index for dataset inhomogeneity
  - data clustering and sequence segmentation without a priori knowledge of the number of clusters or segments
- Application to video summarization based on visual content (key frame extraction)

Clustering

- **Clustering definition: Partition** a given set of objects into (**M)** groups (**clusters**) such that the objects of each group are ‘similar’ and ‘different’ from the objects of the other groups.



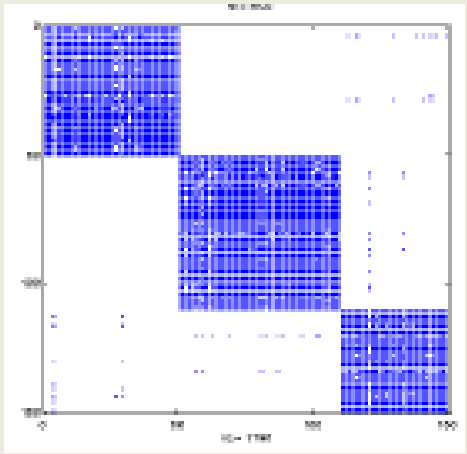
Clustering

- A **distance (or similarity) measure** is necessary
- **Clustering is NP-complete**
- **Unsupervised learning**: no class labels → Difficult to evaluate solutions
- Deciding on the **number of clusters is often difficult**

Clustering

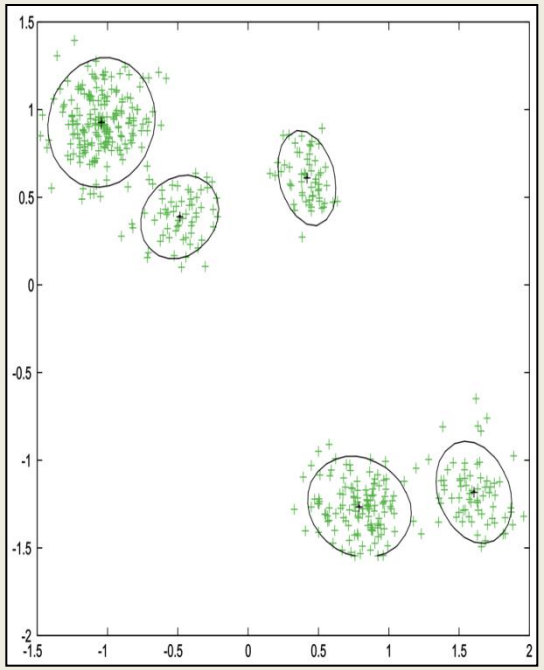
- Clustering Methods
  - Hierarchical (agglomerative, divisive)
  - Density-based (non-parametric, eg DBSCAN)
  - Parametric/model-based (k-means, mixture models, fuzzy c-means etc)
  - Graph-Based

- Clustering Inputs
  - Data Vectors
  - Similarity/Distance Matrix



k-means

- Partition a dataset *X* of *N* vectors x_i into M subsets (clusters) C_k such that **intra-cluster variance is minimized**.
- Intra-cluster variance: sum of distances from the cluster prototype $\boldsymbol{m}_k$



- **k-means**: Prototype = cluster center
- Finds **local minima** w.r.t. **clustering error**

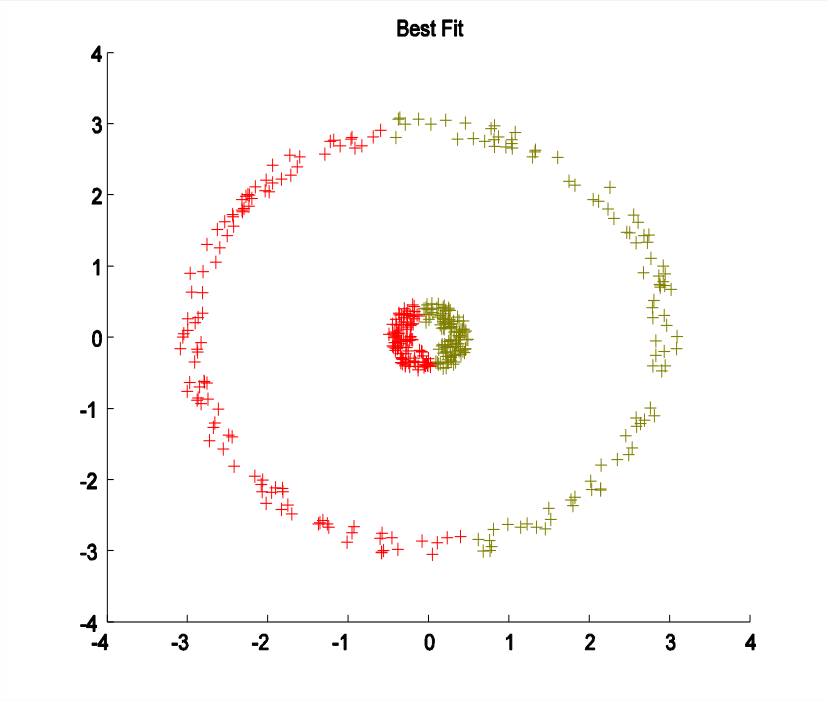
$$E(y) = \sum_{i=1}^{N}\sum_{k=1}^{M} y_{ik} \parallel x_i - m_k \parallel^2,\ y_{ik} = I(x_i \in C_k)$$

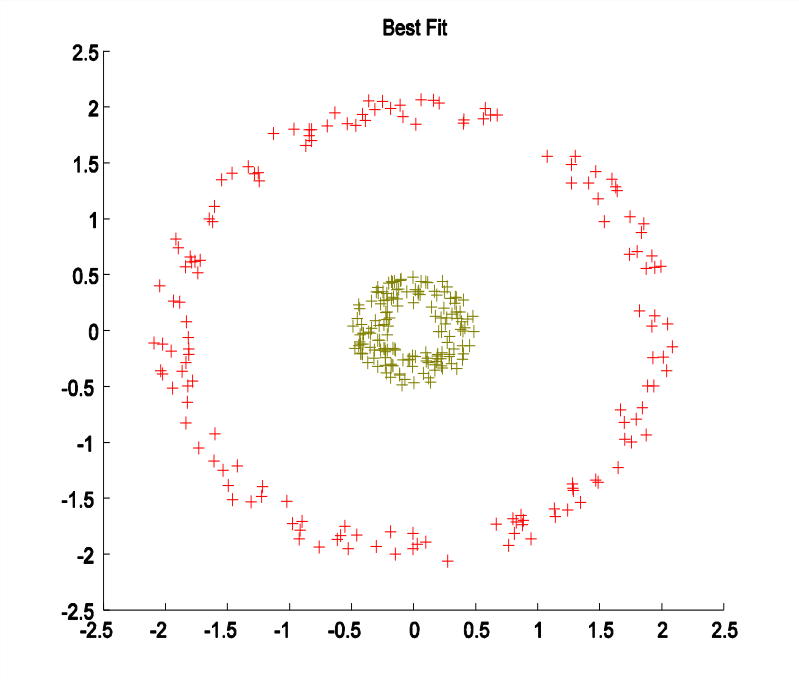
  - sum of intra-cluster variances

- Requires computation of $\parallel x_i - m_k \parallel^2$

Kernel-Based Clustering

2 rings dataset



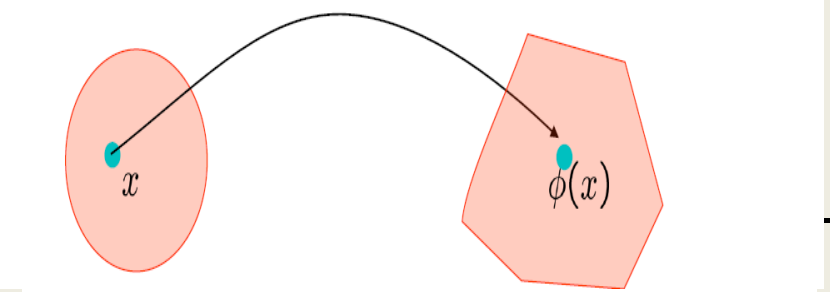


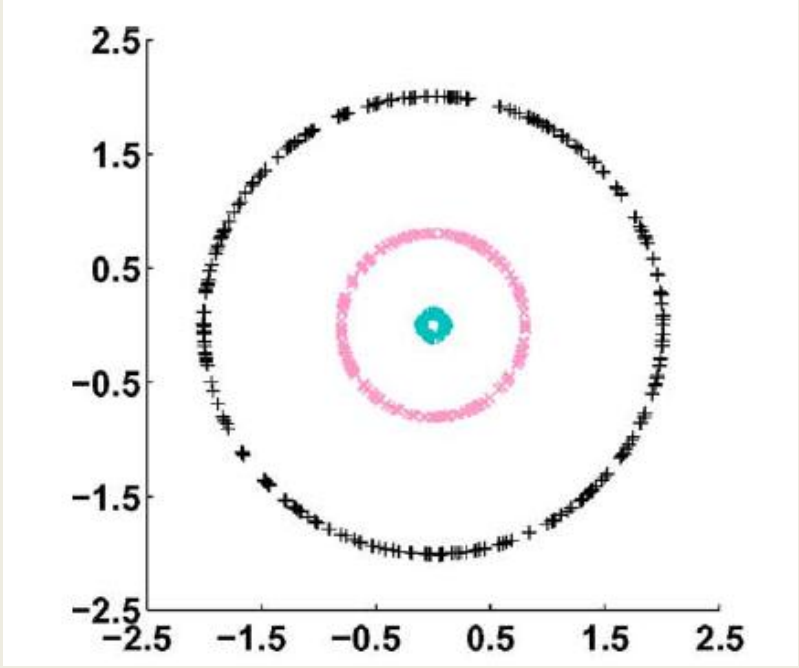
k-means

kernel k-means (RBF kernel,σ=1)

- **k-means** implements **linear** cluster separation

Kernel-Based Clustering
(non-linear cluster separation)





– Given a set of objects and the **kernel matrix** $K=[K_{ij}]$ containing the **similarities** between pairs of objects

– **Kernel trick assumption:** Data points are **mapped** from input space to a higher dimensional **feature space** through a transformation $\varphi(\mathbf{x})$.

RBF kernel:

$K(x,y)=\exp(-||x-y||^2/\sigma^2)$

$$K_{ij}=\phi^T(\mathbf{x}_i)\phi(\mathbf{x}_j) \Rightarrow \|\phi(\mathbf{x}_i)-\phi(\mathbf{x}_j)\|^2=K_{ii}+K_{jj}-2K_{ij}$$

Kernel *k*-Means

- **Kernel k-means = k-means in feature space**
  - Minimizes the **k-means clustering error in feature space**

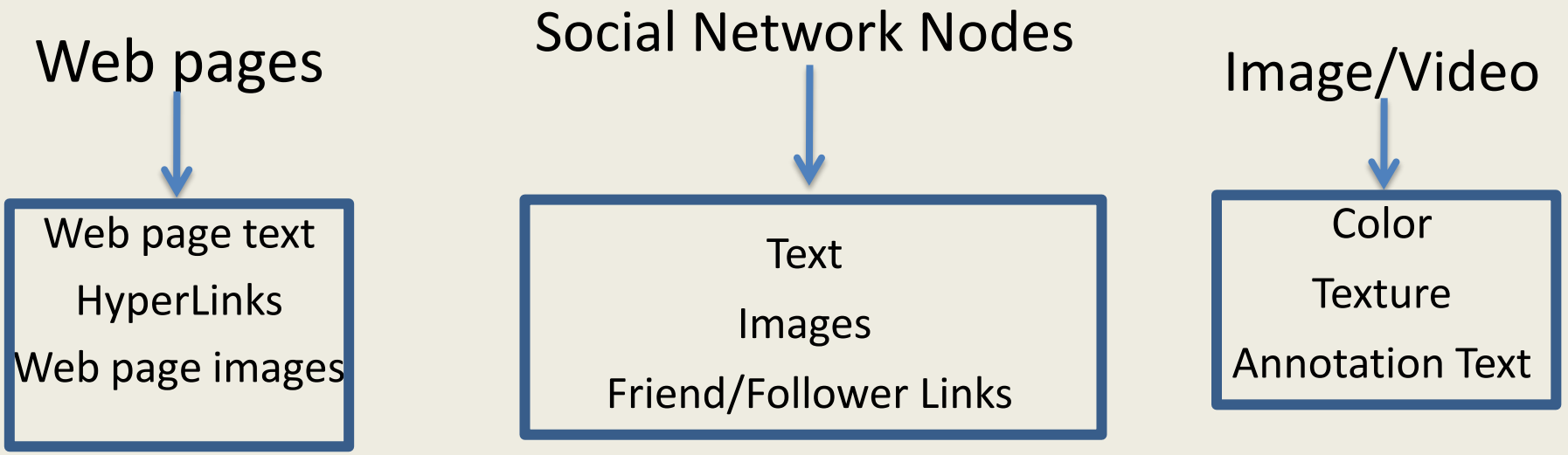
$$E(\boldsymbol{m}_1, \ldots, \boldsymbol{m}_M) = \sum_{i=1}^{N} \sum_{k=1}^{M} I(\boldsymbol{x}_i \in C_k) \|\phi(\boldsymbol{x}_i) - \boldsymbol{m}_k\|^2 \quad \text{where } \boldsymbol{m}_k = \frac{\sum_{i=1}^{N} I(\boldsymbol{x}_i \in C_k)\phi(\boldsymbol{x}_i)}{\sum_{i=1}^{N} I(\boldsymbol{x}_i \in C_k)}$$

- Differences from k-means
  - Cluster centers $\mathbf{m}_k$ in **feature space cannot** be computed
  - Each cluster C_k is explicitly described by its data objects
  - Computation of distances from centers in **feature space**:

$$\|\phi(\boldsymbol{x}_i) - \boldsymbol{m}_k\|^2 = K_{ii} - \frac{2\sum_{j=1}^{N} I(\boldsymbol{x}_j \in C_k) K_{ij}}{\sum_{j=1}^{N} I(\boldsymbol{x}_j \in C_k)} + \frac{\sum_{j=1}^{N}\sum_{l=1}^{N} I(\boldsymbol{x}_j \in C_k) I(\boldsymbol{x}_l \in C_k) K_{jl}}{\sum_{j=1}^{N}\sum_{l=1}^{N} I(\boldsymbol{x}_j \in C_k) I(\boldsymbol{x}_l \in C_k)}$$

Clustering Multimodal Data

Multimodal Data



- **Several representations ('views') of the same data object**

- Each view: either the data vectors or a distance/similarity matrix

Multimodal clustering

- **Multimodal dataset** *X* with N examples and V views:

$$X = \{\mathbf{x}_1, ..., \mathbf{x}_N\}, \quad \mathbf{x}_i = (x_i^{(1)}, ..., x_i^{(V)}), \; x_i^{(v)} \in R^{d_v}$$

- Partition a **multimodal dataset** into *M* disjoint homogeneous groups by taking into account every view
- **Early fusion**: combine views **before** clustering
- **Late fusion**: **aggregate clustering solutions** from individual views
- A better approach: **weighted fusion**: automatically estimate the relevance (weight) of each view **during clustering**
  - views participate in the solution according to their quality (clustering error)
  - 'irrelevant' views are assigned low weight and do not affect the clustering solution

Multi-view Kernel *k*-means(MVKKM)

- Weighted multi-view extension of kernel k-means (Tzortzis and Likas, ICDM'12)
- Assume a **multimodal dataset** *X* with N examples and V views
- For each view: a kernel $K^{(v)}$ → feature transformation $\phi^{(v)}$
- $\boldsymbol{\theta_v}$ the weight of view v (parameters)
- We define a **composite Kernel** K_θ :

$$K_\theta = \sum_{v=1}^{V} \theta_v^p K^{(v)}, \quad \theta_v \geq 0, \ \sum_{v=1}^{V} \theta_v = 1, \ p \geq 1$$

- **p** exponent (user defined)

Multi-view Kernel *k*-means(MVKKM)

- Find partition **y** (in M clusters) and view weights **θ** that **minimize** kernel k-means error for kernel matrix $\mathbf{K}_{\boldsymbol{\theta}}$

$$E(\theta, y) = \sum_{i=1}^{N}\sum_{k=1}^{M} y_{ik} \| \phi_\theta(x_i) - m_k \|^2, \; y_{ik} = I(x_i \in C_k)$$

$$E(\theta, y) = \sum_{v=1}^{V} \theta_v^p D_v(y), \; \theta_v \geq 0, \; \sum_{v=1}^{V} \theta_v = 1$$

$$D_v(y) = \sum_{i=1}^{N}\sum_{k=1}^{M} y_{ik} \| \phi_v(x_i^v) - m_k^v \|^2, \; y_{ik} = I(x_i \in C_k)$$

Kernel k-means clustering error of y for view v

Multi-view Kernel *k*-means(MVKKM)

- Starting from some initial θ_v (usually 1/V) and an initial clustering y, **perform the following two steps until convergence**:
  - **update** the cluster assignments **y** using **kernel k-means** with current K_θ
  - **update θ_v** :

$$\theta_v = 1 / \sum_{v'=1}^{V} \left(\frac{D_v(y)}{D_{v'}(y)} \right)^{1/(p-1)} \quad \mathbf{p > 1}$$

$$\theta_v = 1,\ v = \arg\min D_{v'}\ \ \theta_v = 0,\ v \neq \arg\min D_{v'},\ \mathbf{p = 1}$$

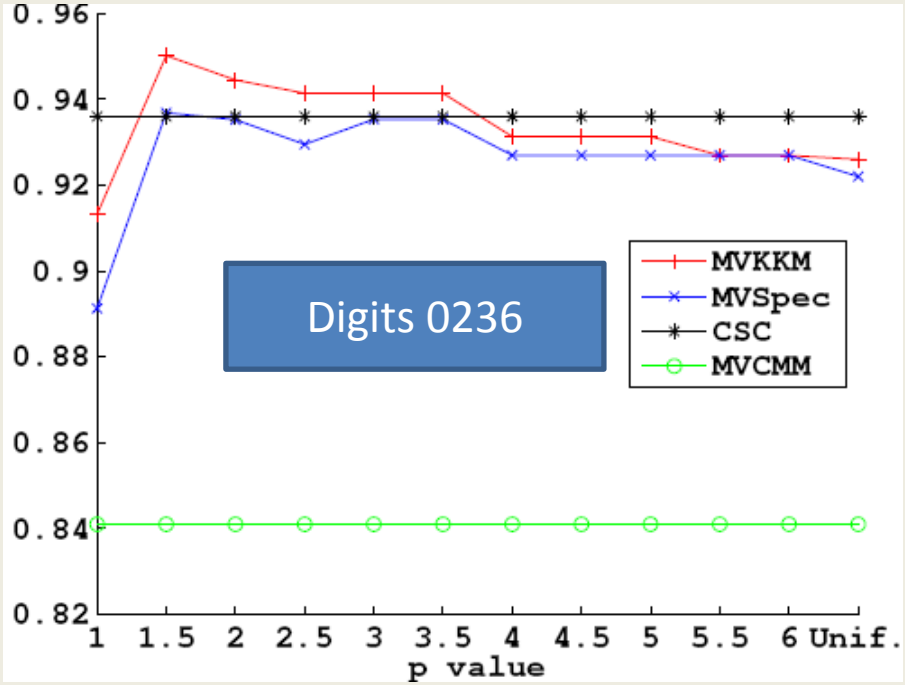
- Views with lower clustering error are assigned higher weight
- **Feature space changes** at each clustering iteration; however convergence is guaranteed
- p regulates **weight sparsity** (typical value p=2)

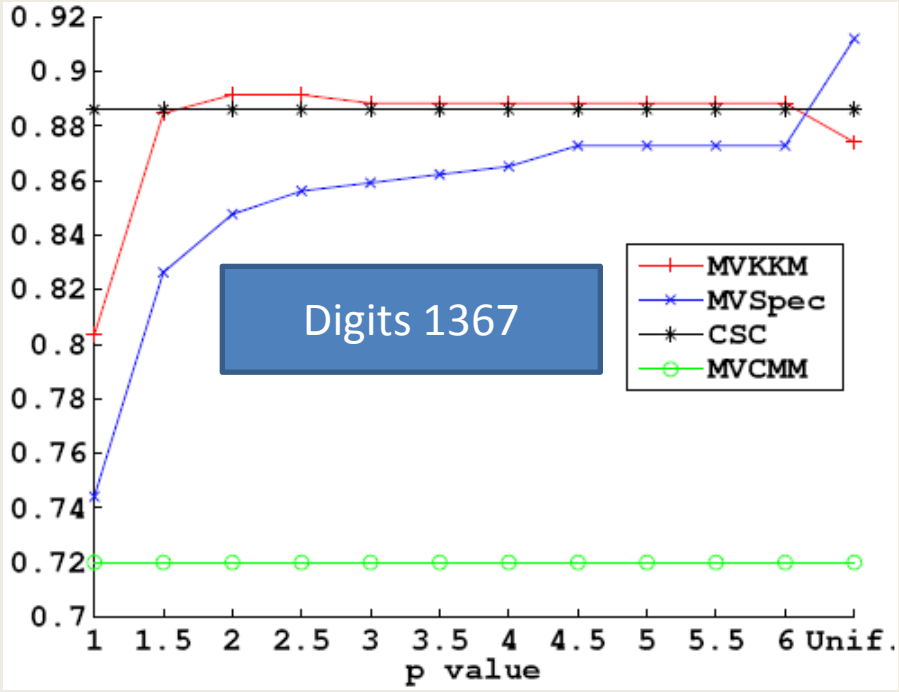
Experiments – Real Datasets

- *Multiple Features* – Collection of handwritten digits
  - Five views
  - Ten classes
  - 200 instances per class
  - Extracted several four class subsets

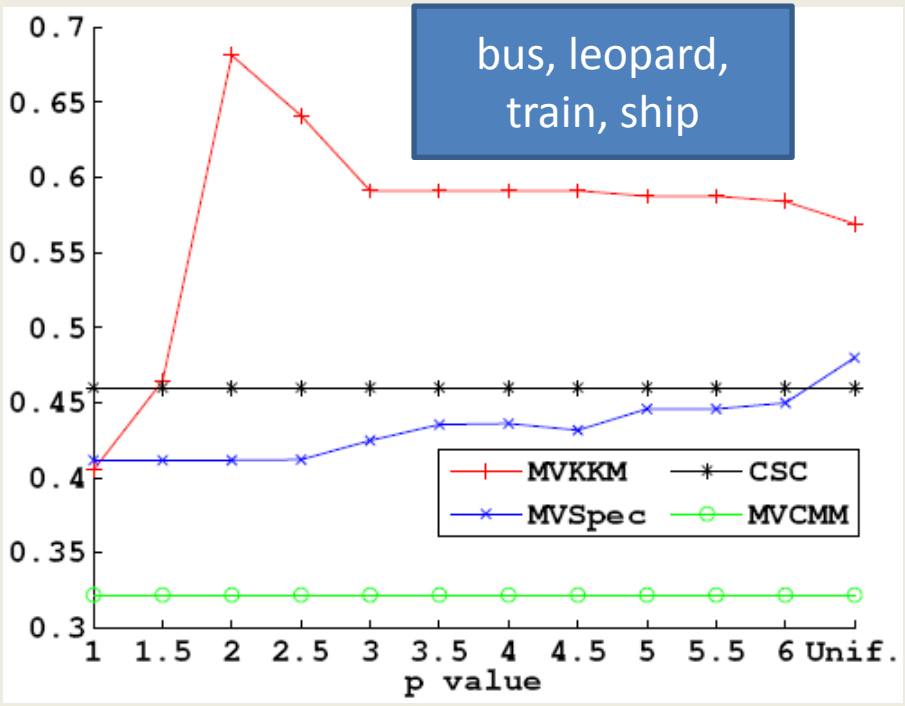
- *Corel* – Image collection
  - Seven views (color and texture)
  - 34 classes
  - 100 instances per class
  - Extracted several four class subsets

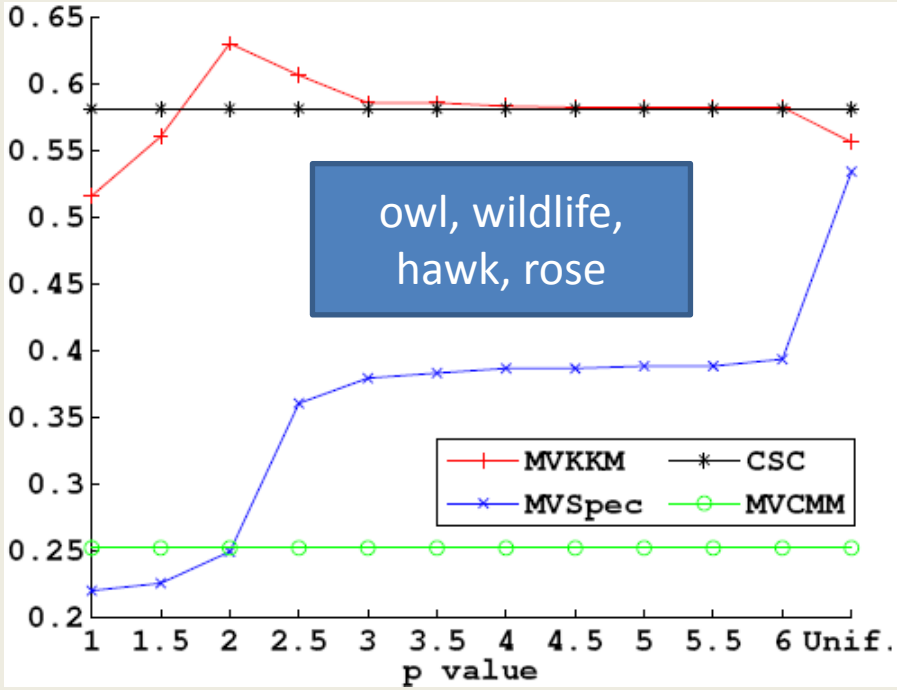
Experiments – Digits





Experiments – Corel





MVKMM provides best results for p=2

Key-frame Extraction

- **Video summarization** based on **visual content**
- Usually a **video sequence** is **decomposed into shots**.
- A shot is defined as an unbroken sequence of video frames taken from a single camera.
- **Keyframes**: summarization of a video shot
  - **rapid assessment** of the video content by inspecting the key-frames of the shot
  - define **shot similarity**: scene segmentation, content-based retrieval, rushes summarization.
- Keyframes should represent the whole video content without missing important information.
- Keyframes should not be similar, in terms of video content information.




















Key-frame Extraction Approaches

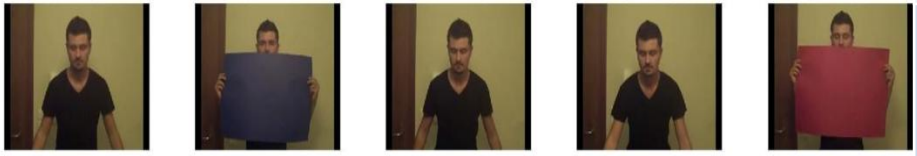
- **Clustering**-based (e.g. k-means, spectral approaches)
  - The medoid frame of each group of frames is selected as key-frame

- Frames are represented as feature vectors: **image descriptors** based on color, texture, interest points
- A **single image descriptor** does not suffice for all cases.
- **Weighted Fusion** of different image descriptors (e.g. color, SIFT etc) (Ioannidis, Chasanis and Likas, ICPR'14, PRL'16)

Image Descriptors

- **HSV Color Histograms**:
    - HSV1D (96 bins -> 64H + 16S + 16V)
    - HSV3D (128 bins-> 8H x 4S x 4V)
- **Census Transform Histogram** (CENTRIST) (251 bins)
- **Wavelet**:
  - 9 Haar wavelet sub-bands are used on 3x3 grids to form a 81-d feature vector.
- **SIFT (20 or 50 visual words)**

Example (V=2)

a)		***GROUND TRUTH (M=5)***
b)		*HSV*
c)		*CENTRIST*
d)		*WAVELET*
e)		*HSV+CENTRIST*
f)		*HSV+WAVELET*
g)		*CENTRIST+WAVELET*

Weights

HSV	*Centrist*	*Wavelet*
0.9880	0.0120	-
0.9449	-	0.0551
-	0.3286	0.6714

Performance Evaluation

- No single descriptor dominates the other single descriptors -> **fusion necessary**

- Regardless of the pair of descriptors, **the weighted fusion of two descriptors always provides better or equal performance** than
  - using each descriptor individually or
  - using the unweighted combination of the two descriptors

Detection Accuracy

Descriptors	Incremental single view	Spectral single view
HSV1D	63.08	61.90
HSV3D	64.52	69.28
SIFT20	61.96	71.28
SIFT50	61.16	69.65
CEN	62.98	68.41
WAV	66.03	66.47
	Unweighted-multiview	**Weighted multiview**
HSV1D-SIFT20	73.17	**78.82**
HSV1D-SIFT50	72.70	**79.44**
HSV3D-SIFT20	74.25	**83.26**
HSV3D-SIFT50	73.94	**83.88**
HSV1D-CEN	70.98	**83.26**
HSV1D-WAV	69.74	**81.72**
HSV3D-CEN	69.74	**80.79**
HSV3D-WAV	70.42	**82.03**

Multimodality
for number of clusters estimation

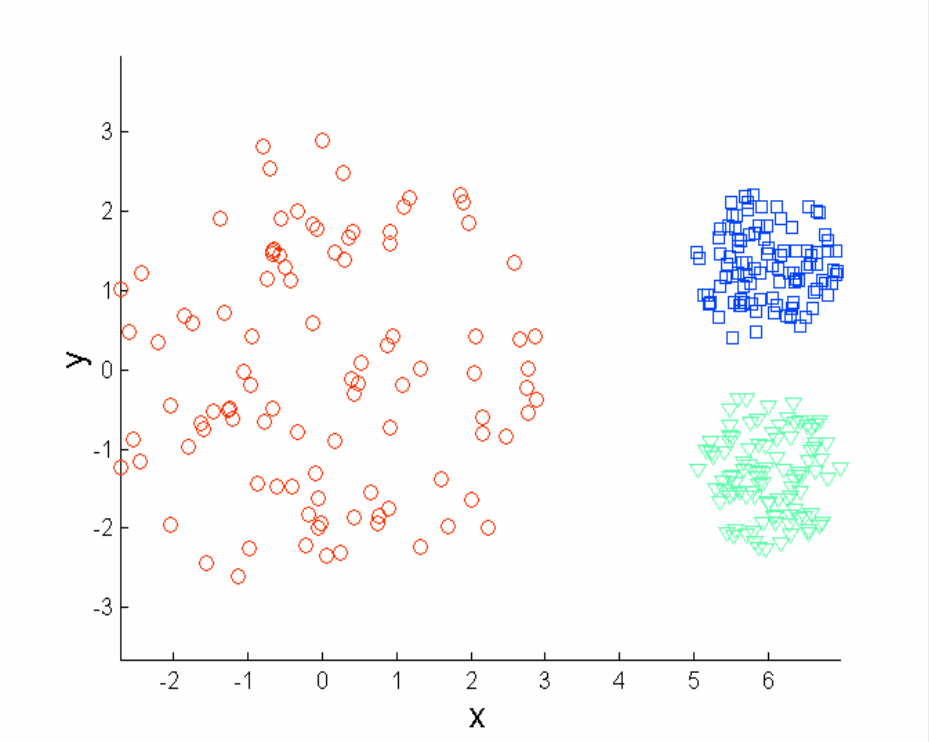
Deciding on the # of clusters

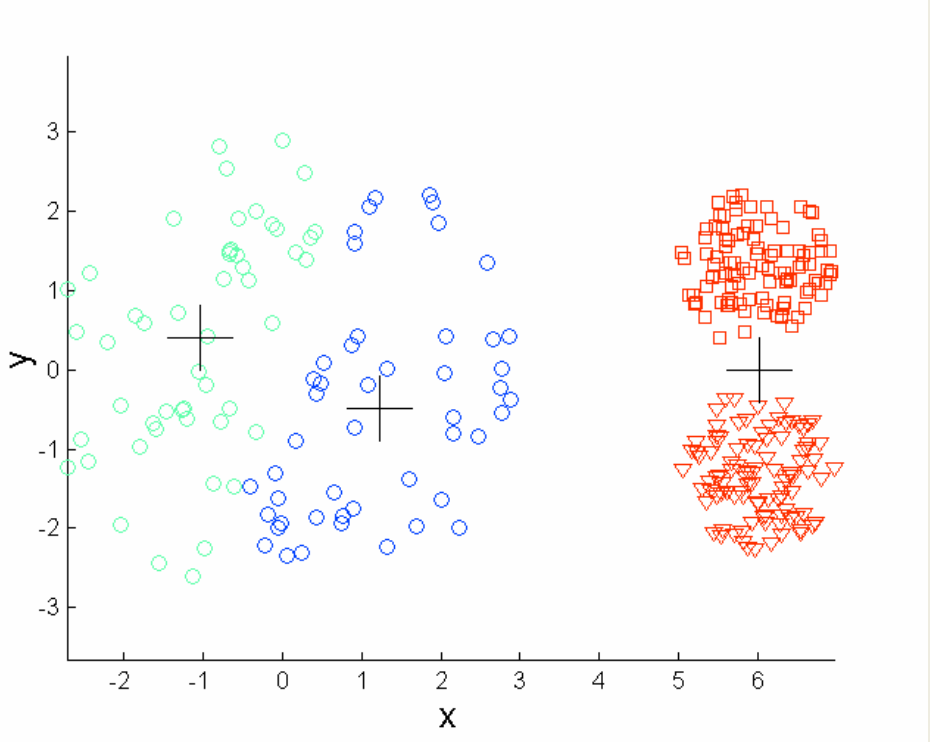
- **Selection Approaches**: Use a **Criterion** to **select** among the solutions for several values of M (kmeans or GMMs are used)
  - Clustering Objective(M) + Model Complexity(M) (BIC, MML)
  - Marginal Likelihood (Bayesian GMMs)
  - Gap Statistic
  - Variance Ratio Criterion (**VRC**):

    (intracluster variance)/ (intercluster variance)

Deciding on the # of clusters

- Optimal solutions wrt clustering error **do not** always reveal the true clustering structure





Deciding on the # of clusters

- The critical issue:

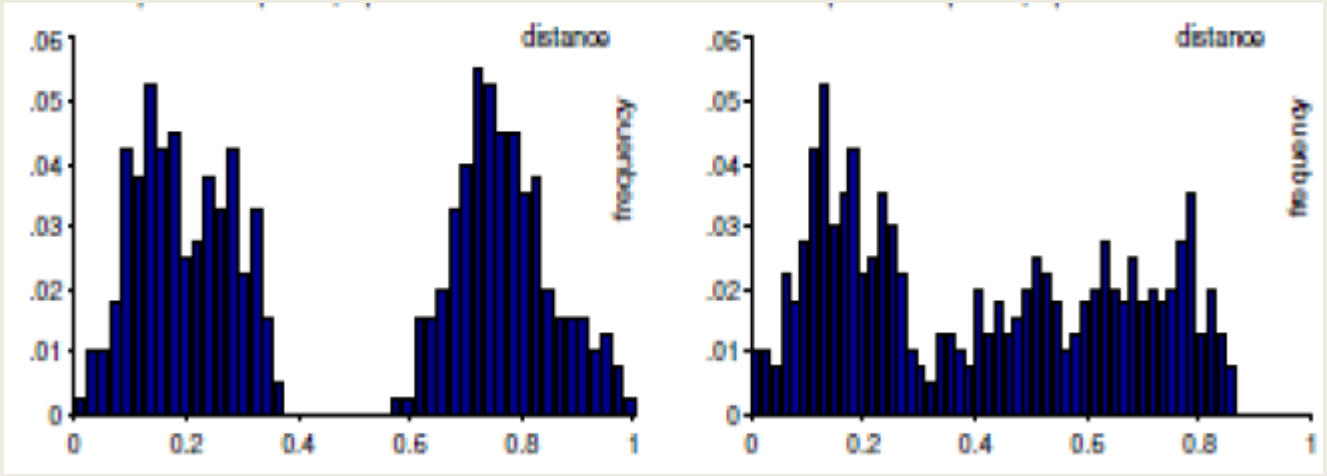
  deciding on the content homogeneity of a set of data objects given either the data vectors or the similarity matrix.

Deciding on dataset homogeneity

- One approach: **split the set in two subsets**, apply a criterion (e.g. BIC) for M=1 and M=2 and decide if there is improvement or not (**x-means** algorithm)
- Another approach: Test for **Gaussianity** (if data vectors available):
  - Project the data in the principal direction and apply 1-d Gaussianity test (**g-means** algorithm)
- Our approach: **tests for unimodality** applied to the **similarity matrix** (does not require the data vectors)

Dip test for unimodality

- Given **a set X of real numbers**
- **Hartigans' dip-test for unimodality** (Annals of Statistics, 1984)
  - computes **dip measure**: "departure from unimodality" of the empirical distribution (histogram) of **X**
- **Dip(X)**: distance of the empirical data distribution of X from the closest unimodal distribution

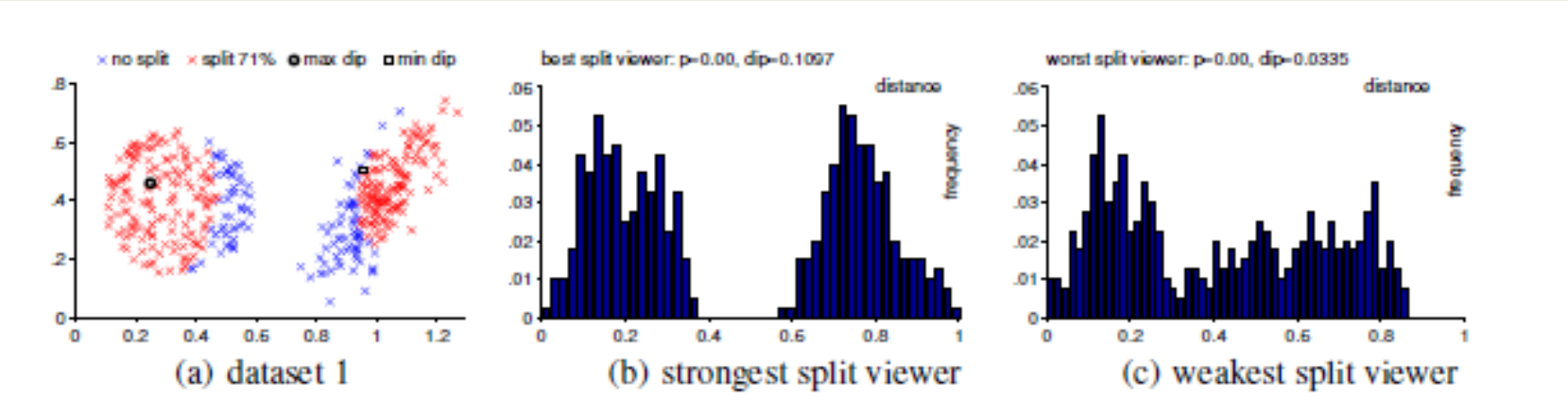


dip=0.11 dip=0.03

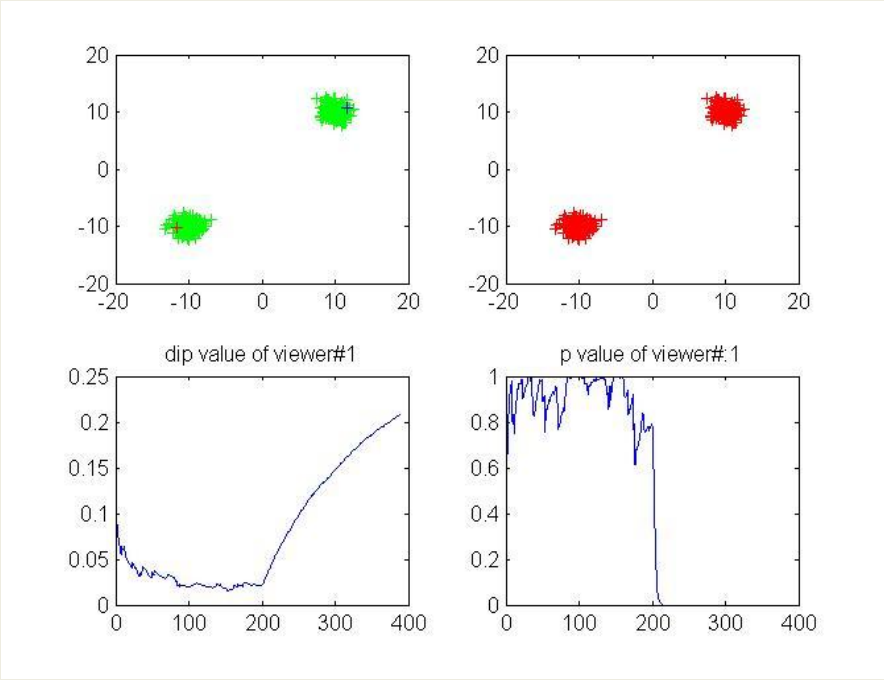
Dip test for unimodality

- **Uniform** is considered as the **‘extreme’** unimodal distribution
- The **statistical significance** (**pval**) of a dip value is assessed through bootstrapping (off-line)
- pval=0 → multimodality, pval=1→ unimodality

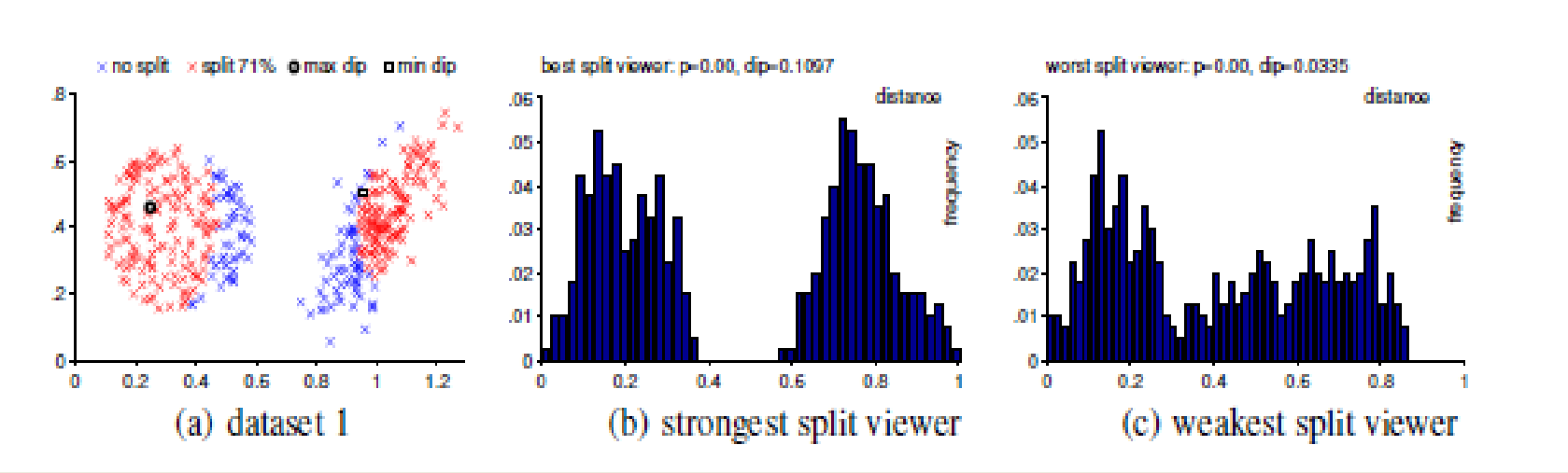
Dataset Multimodality: Dip-dist criterion (Kalogeratos & Likas NIPS'12)



- Reference object ('viewer')
- The **histogram of its distances** of all the dataset objects is checked for multimodality (dip test)
- Viewer detecting multimodality → '**split viewer**' (multimodal viewer)



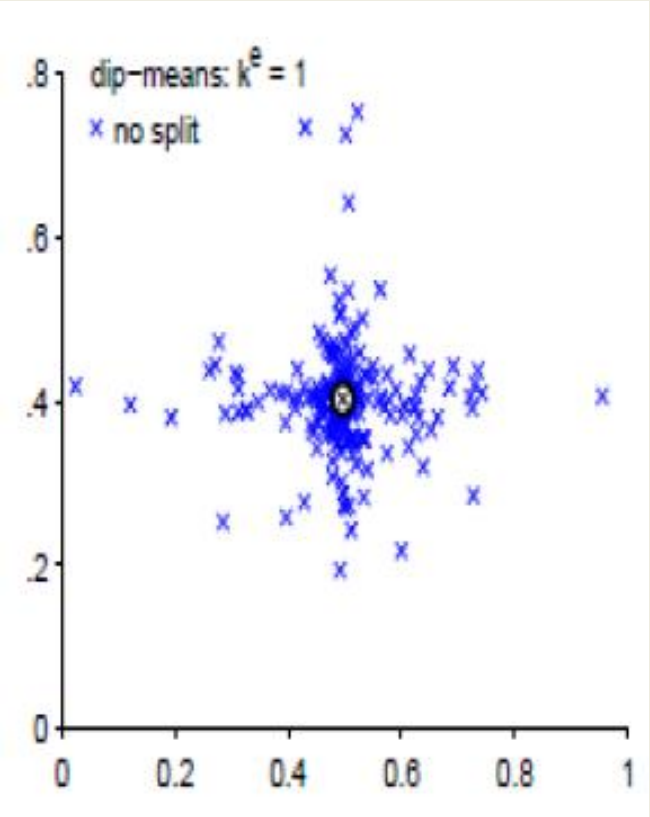
Dataset Multimodality: Dip-dist criterion

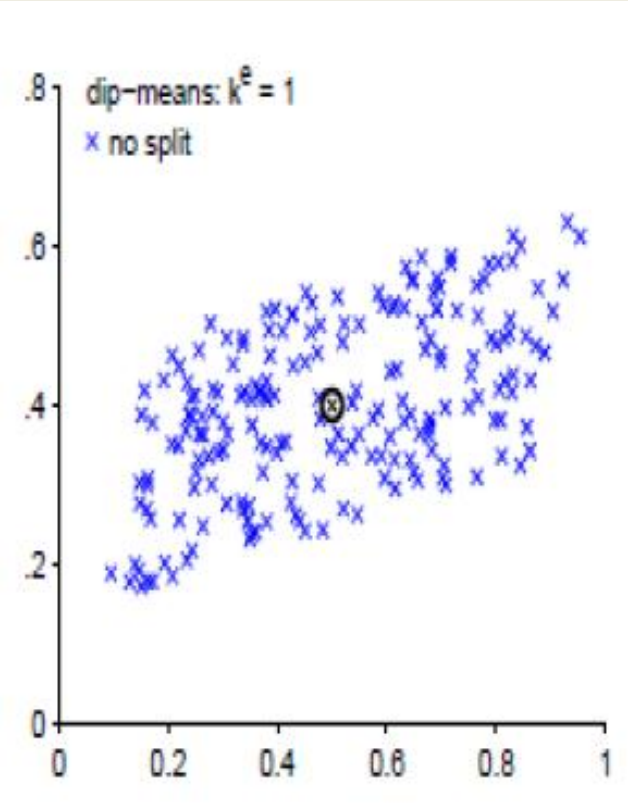


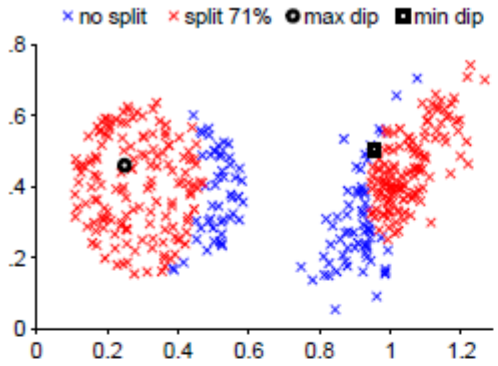
(a) dataset 1 (b) strongest split viewer (c) weakest split viewer

- How to select viewers?
- Each object x_i is used as 'viewer' -> dip-test applied on the elements of row i of the distance (similarity) matrix
- **dip-dist criterion: dip-test** applied on each **rows** of the similarity (distance) matrix
- **Sufficient split viewers** → dataset **multimodal**

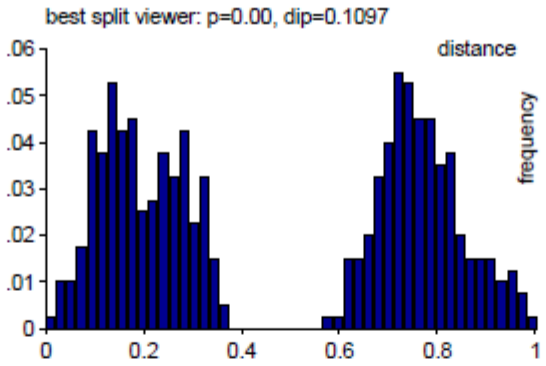
Dip-dist criterion – unimodal examples



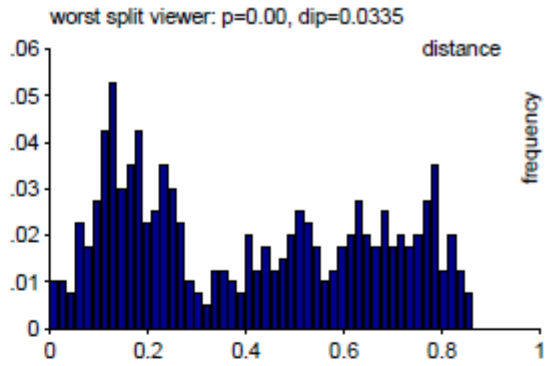




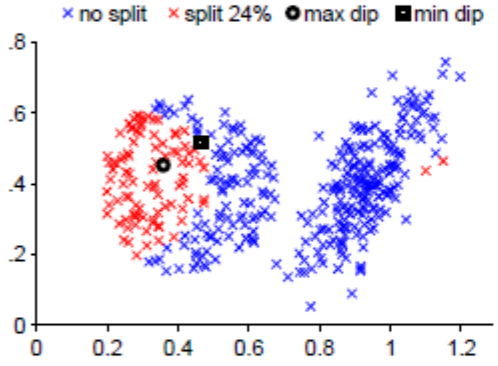
(a) dataset 1



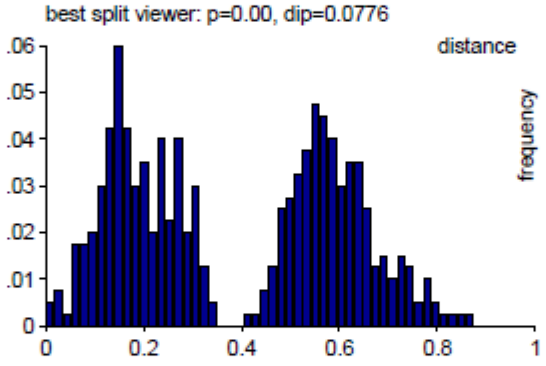
(b) strongest split viewer



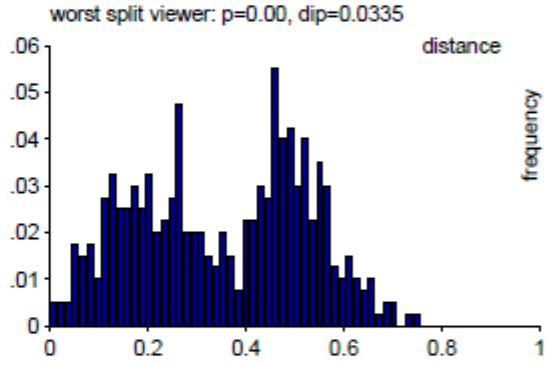
(c) weakest split viewer



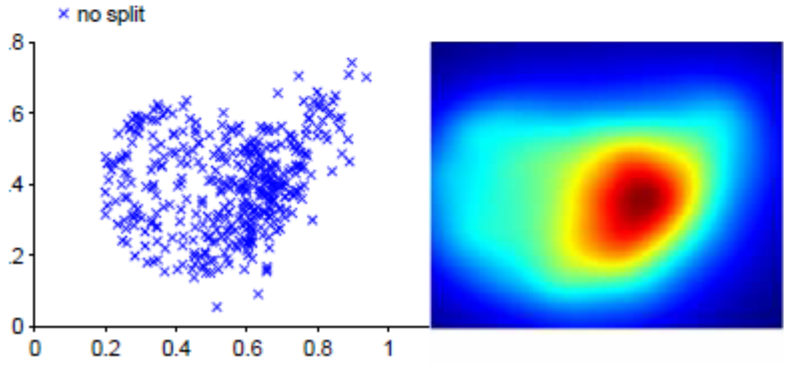
(d) dataset 2



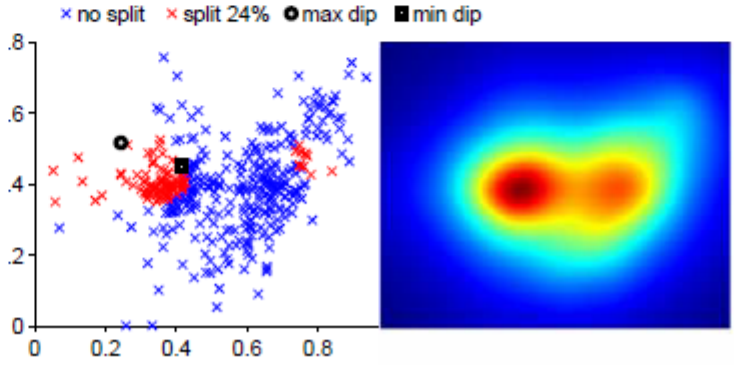
(e) strongest split viewer



(f) weakest split viewer



(g) dataset 3 (h) density plot



(i) dataset 4 (j) density plot

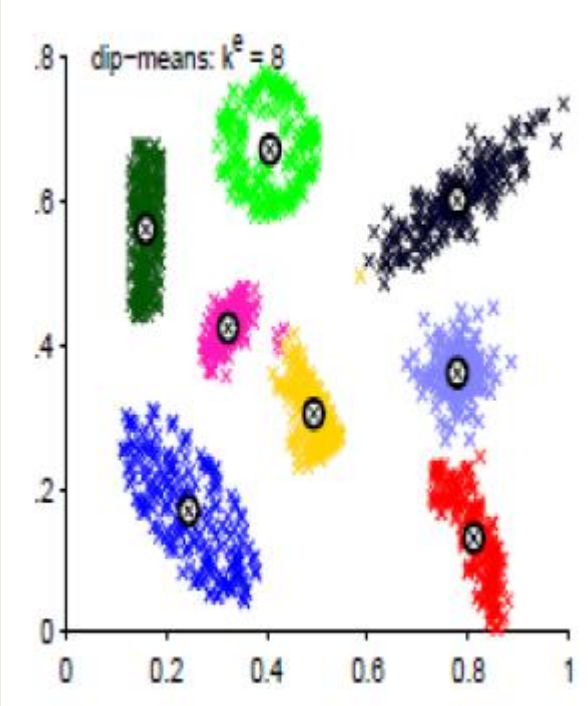
Incremental Dip (Dip-means)

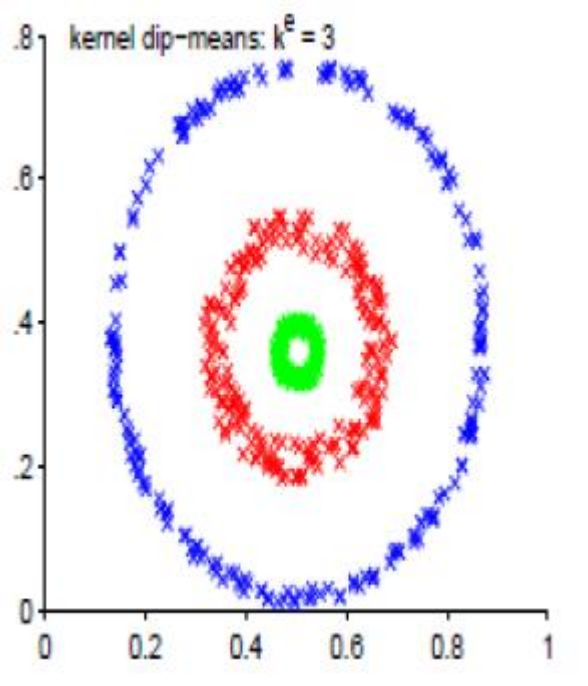
- **Divisive** clustering using **k-means** and cluster **splitting**
- **Determine** and **split multimodal** clusters until all clusters become **unimodal**
- **Dip-means algorithm** (starts with k=1)
  - apply **dip-dist** for each cluster j and find the **multimodal** cluster with **maximum score:**

$$score_j = \frac{1}{|v_j|} \sum_{x \in v_j} dip(dist(x))$$

  - split this cluster in two sub-clusters and run k-means with k+1 clusters. Set k:=k+1
  - until all clusters are unimodal
- **Kernel Dip-means : Kernel k-means** is used instead of k-means

Dip-means examples

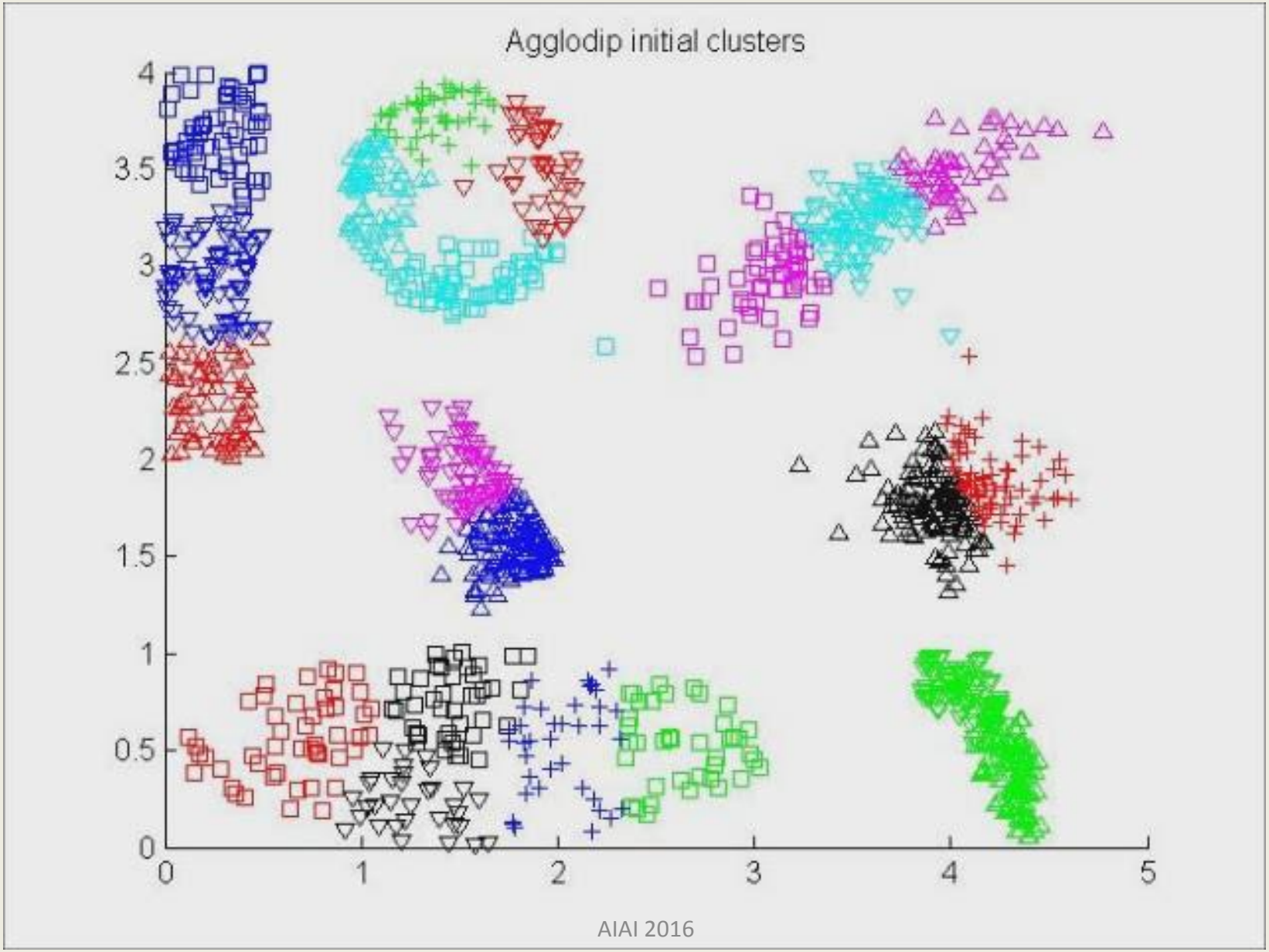




Agglodip (Agglomerative Clustering)

- **Merge test**: merge two datasets and decide on the unimodality of the resulting set
- Starts with an **initial partition** of the dataset into **small unimodal** clusters (e.g. using k-means or kernel k-means)
- At each iteration
  - a **merge-test** is applied to all pairs of clusters in the current solution and the **unimodal pairs** are identified.
  - The **unimodal** pair with the **minimum dip-value (maximum unimodality)** is selected for merging and a new iteration starts.
- Terminate when there does not exist any pair of clusters succeeding in the merge-test.

Aggodip example



Aggodip issues

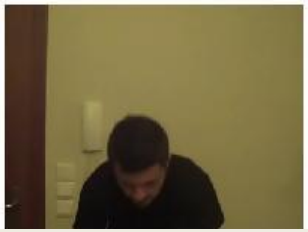
- When merging a **large** cluster with a distinct **small** cluster, it is possible that the merge-test will decide unimodality.
    - Solution: **merge clusters** with **comparable** sizes

- Use the **cluster centroids as viewers** (two viewers only)

- Create a **0-1 neighborhood graph** of the initial clusters based on the merge test and find the **connected components** of this graph

Ioannidis, Chasanis and Likas, ICMV'14

Agglodip

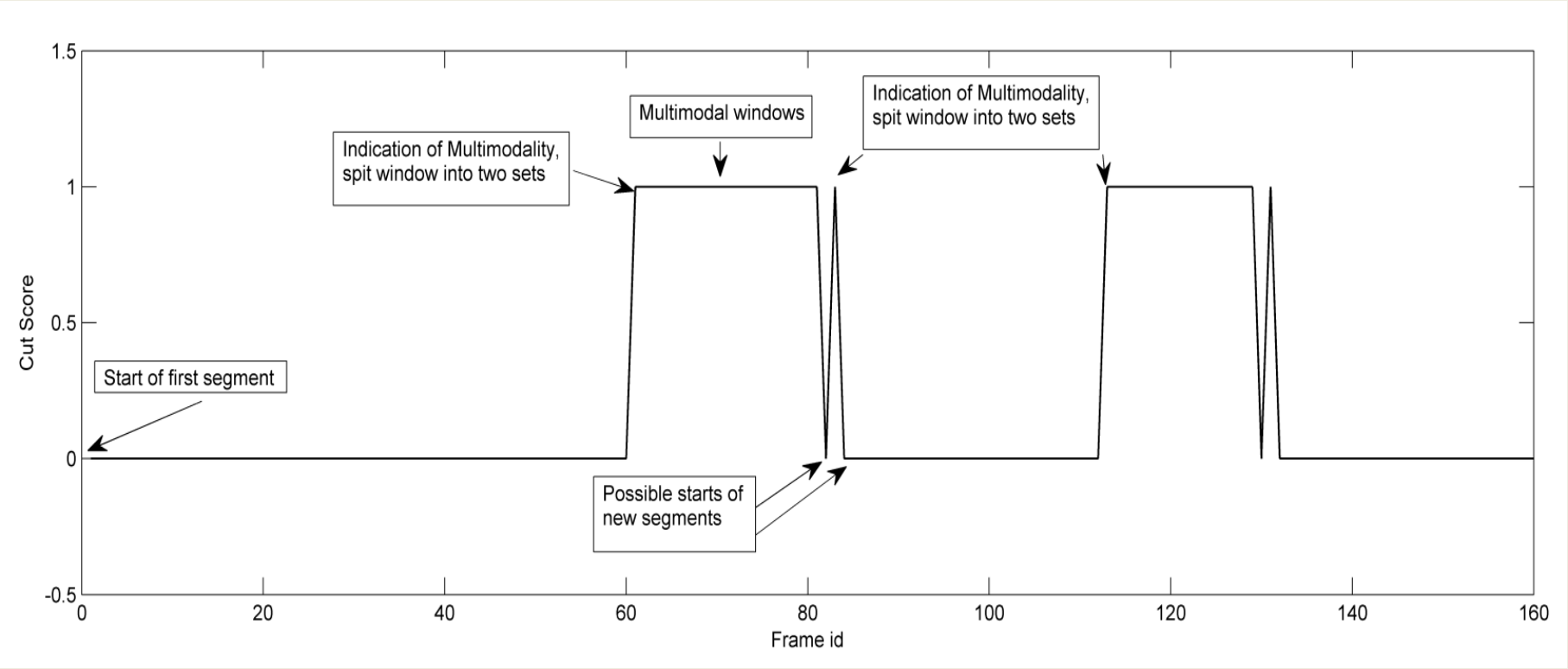

Sequence segmentation (SegmentDip)

- Assume a **sequence** (stream) of **data objects**
- A similarity/distance measure between pairs of data objects is given
- **Segmentation Task**: Partition the sequence into **homogeneous** segments by determining appropriate **cut points (number of segments is not given)**

- **SegmentDip** method
  - **Sliding window of fixed size w** over the sequence
  - Test the **homogeneity of each window** with dip-dist
  - If a **window** is found **multimodal**, a **cut point** may be set in this window.

SegmentDip

Ioannidis, Chasanis , Likas, ICSP'14



- The method is **fast** since it applies dip-dist criterion on **w** data objects each time

Video Segmentation

- **Segmentation into shots**: small window, detects sharp changes in video content

- **Segmentation of each shot (keyframe extraction)**: larger window, detects smoother variations in video content

Video Segmentation Example

VideoSum software

Multimodality in Data Clustering

- **Open problem**: Combining the two notions of multimodality into an algorithm for **multimodal clustering** that estimates **automatically both** the **number of clusters M** and the **weights θ** of each view.

  - Multiview clustering methods assume M is given and estimate kernel matrix K_θ
  - Dip-based methods assume kernel matrix K is given and estimate number of clusters M.

Thank you!