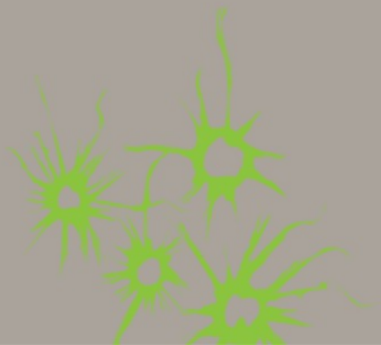
UNIVERSITY OF TWENTE.

UNSUPERVISED LEARNING

Mahdi KHODADADZADEH

October 2021

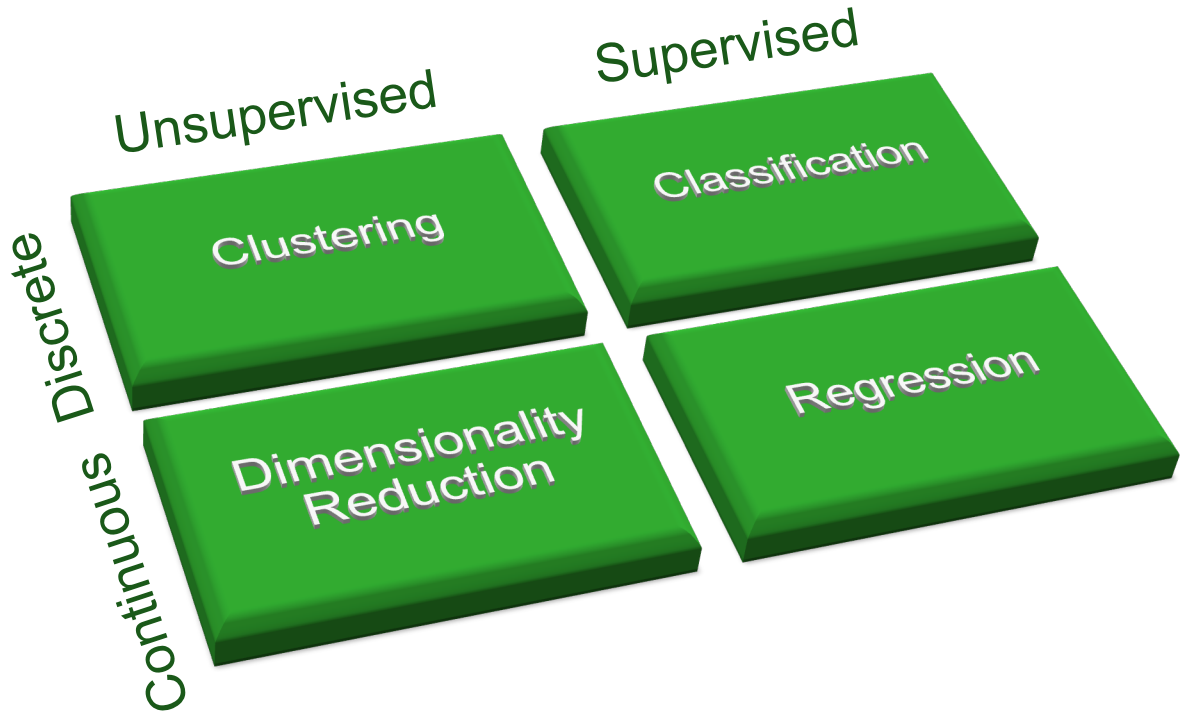


ITC FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

UNSUPERVISED VS SUPERVISED LEARNING

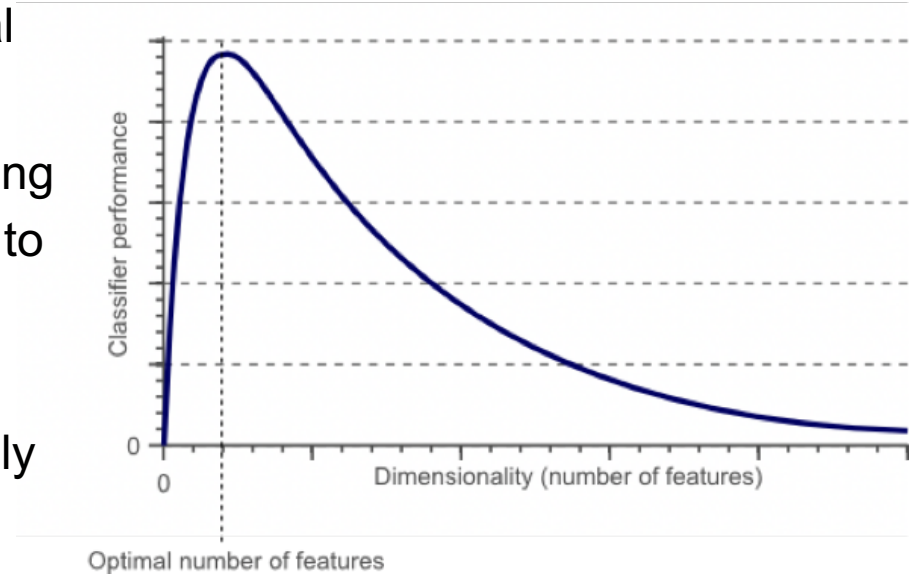
- Unsupervised learning (clustering)
  - Class labels unknown
  - Checking for groups/clusters in the data
- Supervised learning (classification)
  - Supervision: training data to create model
  - Classification of new data

MACHINE LEARNING PROBLEMS



DIMENSIONALITY REDUCTION

- Curse of dimensionality
  - Working with high-dimensional data
  - The difficulties related to training machine learning models due to high dimensional data
  - The number of training data needed increases exponentially with each added feature



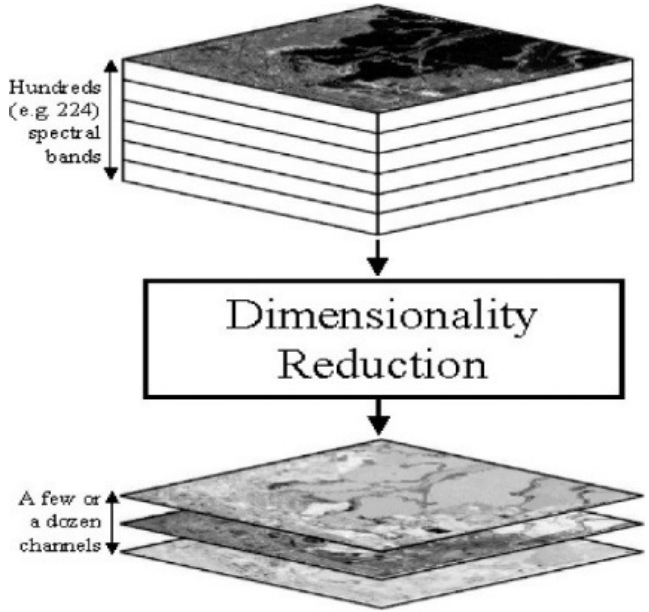
DIMENSIONALITY REDUCTION

- Choosing discriminating and independent features is key to any machine learning algorithm
- In real applications usually many features are measured while only a very small percentage of them carry useful information towards our learning goal

- We usually need an algorithm that compress our feature vector and reduce its dimension

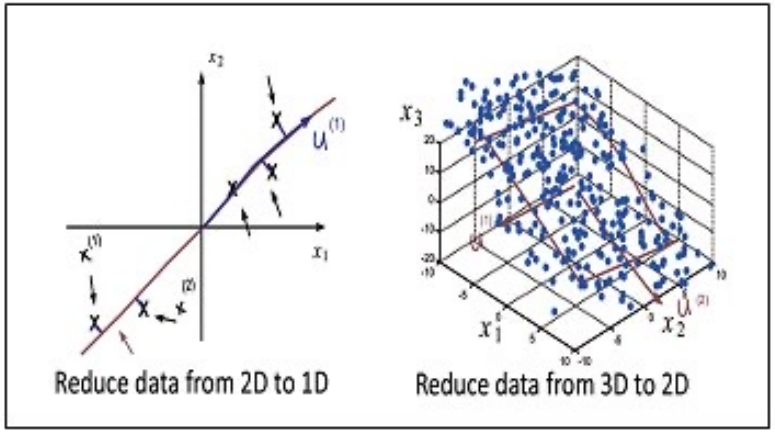
PRINCIPAL COMPONENT ANALYSIS (PCA)

- Relatively simple and popular technique
- PCA converts a set of observations into a set of linearly uncorrelated variables, called principal components
- Represents data in a space that better describes the variation
- If a strong correlation between variables exists, the attempt to reduce the dimensionality is reasonable



PRINCIPAL COMPONENT ANALYSIS (PCA)

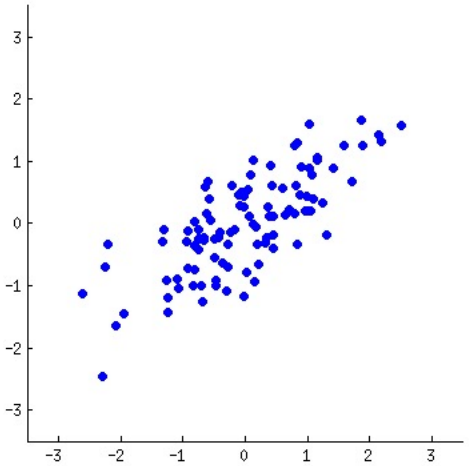
- Identifies directions of maximum variance (in high-dimensional data) and projects the data onto a smaller dimensional subspace while retaining most of the information.
- PCA projects the entire dataset onto a different feature (sub)space

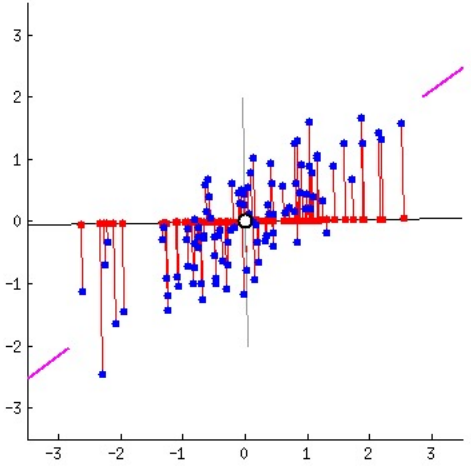


From: Gosavi, V., A. Deshmane, and G. Sable. "Adaptive Neuro Fuzzy Inference System for Facial Recognition." *IOSR Journal of Electrical and Electronics Engineering* 14.3 (2019): 15-22.

PRINCIPAL COMPONENT ANALYSIS (PCA)

- What the projections look like for different lines (red dots are projections of the blue dots)
- The reconstruction error are given by the length of the connecting red line





From: https://stats.stackexchange.com/questions/2691/making-sense-of-principal-component-analysis-eigenvectors-eigenvalues

PRINCIPAL COMPONENT ANALYSIS (PCA)

- PCA is built on the concepts of Eigenvector & Eigenvalues
  - What is an eigenvector?

    Eigenvectors are those vectors that their direction does not change when a linear transformation (such as multiplying it to a scalar) is performed to them.
  - What is an eigenvalue?

    The scalar that is used to transform (stretch) an Eigenvector.

constructing one vector with one value to represent the covariance matrix

A * *x* – Lamda * *x* = 0

A = Covariance Matrix

PRINCIPAL COMPONENT ANALYSIS (PCA)

→ Compute eigenvectors (i.e. the principal components) of a dataset

→ Organize them into a matrix.

→ Each eigenvectors is associated with an eigenvalue (i.e. the “length” or “magnitude” of the eigenvector).

→ If some eigenvalues >> than others, then the reduction of the dataset via PCA onto a smaller dimensional subspace is reasonable.

→ This is done by dropping the “less informative” eigenpairs

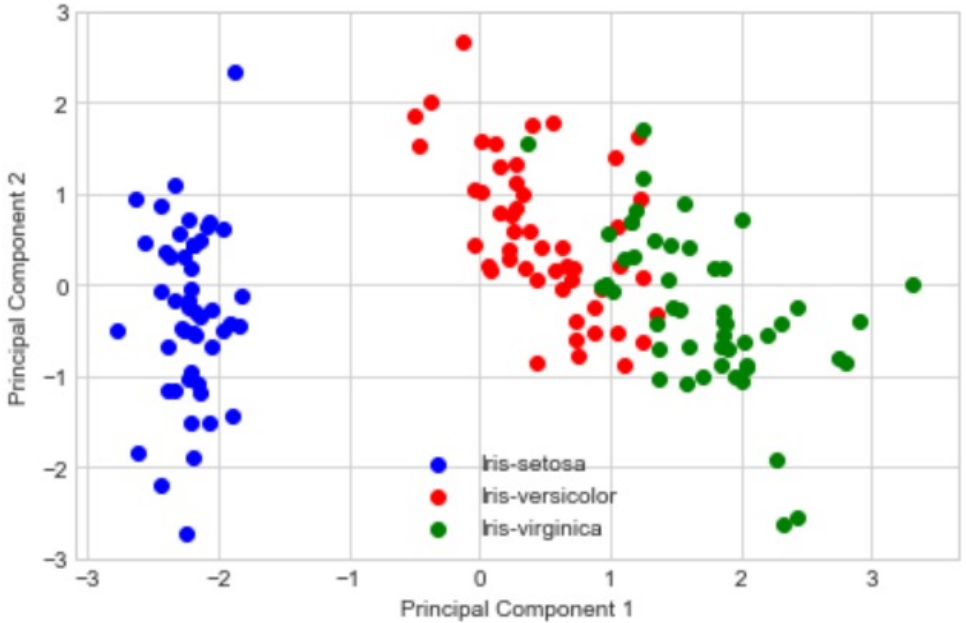
PRINCIPAL COMPONENT ANALYSIS (PCA)

Standardize the data

1. Obtain the Eigenvectors and Eigenvalues from the covariance matrix or correlation matrix
2. Sort eigenvalues and pick the k eigenvectors that correspond to the k largest eigenvalues [k is the number of dimensions of the new feature subspace ($k \leq d$)]
3. Create the projection matrix W of the selected k eigenvectors.
4. Transform the original dataset X via W to obtain a k-dimensional feature subspace Y

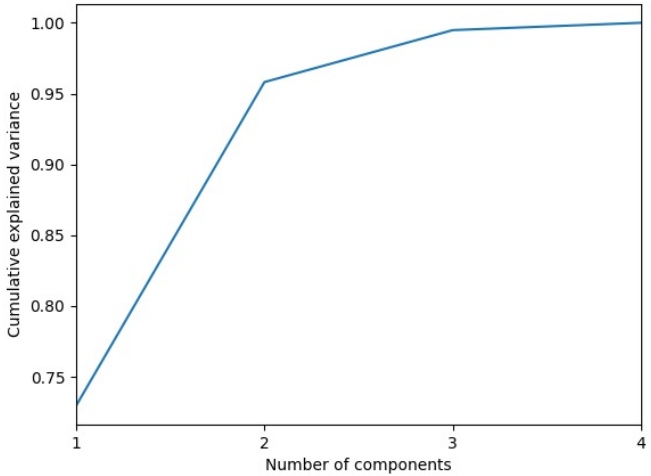
PRINCIPAL COMPONENT ANALYSIS (PCA)

- Python → sklearn.decomposition.PCA



PRINCIPAL COMPONENT ANALYSIS (PCA)

- How many PCs?
  - One result of PCA is a measure of the variance corresponding to each PC relative to the total variance of the dataset
  - From that we can calculate the percentage of variance explained for the m-th PC.



WHY CLUSTERING?

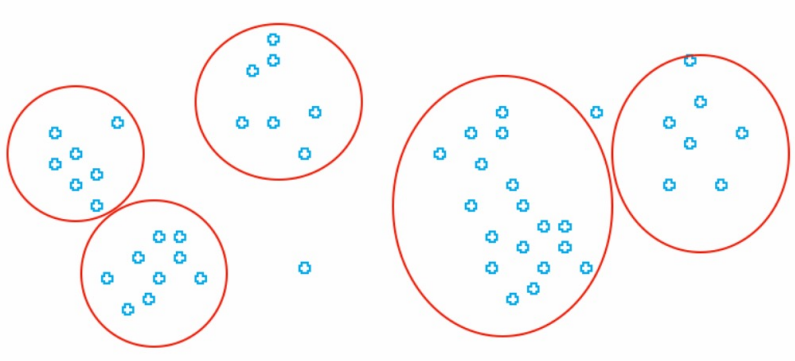
- Large amounts of spatio-temporal data becoming available both to the scientific community and to the general public.

"Data is arguably the most important natural resource of this century… Big data is big news just about everything you go these days. Here in Texas everything is big so we just call it data"

Michael Dell, 2014

CLUSTERING

- An important task in data mining that aims at identifying groups of elements that are similar among themselves but dissimilar to the elements in other groups.
- It provides a high-level abstraction of the data, which facilitates the extraction of useful information

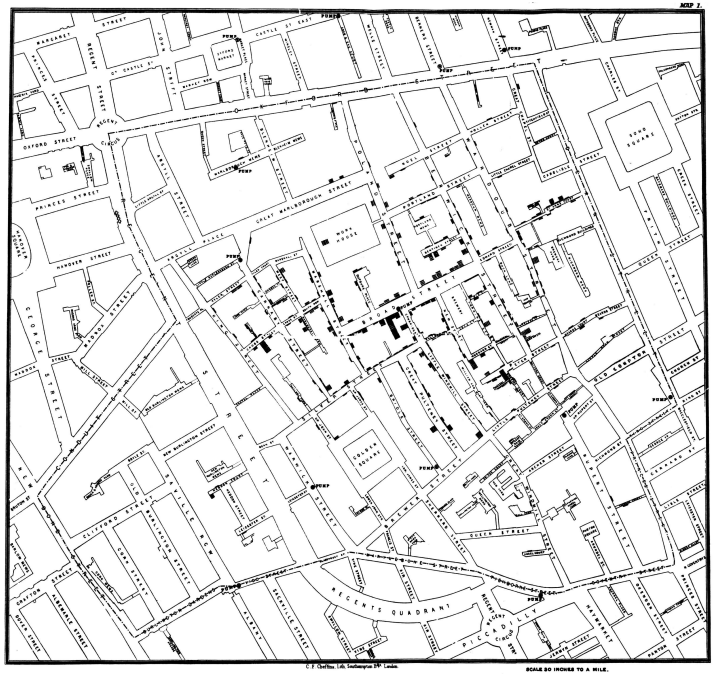


CLUSTERING

- Contrarily to classification/regression task there is NOT a target
- It is suitable in problems where we have unlabeled objects or where the process of labelling is expensive (time / money).
- It is often done as part of the EDA to get a better grip on the objects at hand

HISTORIC APPLICATION OF CLUSTERING

- John Snow a London physician plotted the location of cholera deaths on a map during an outbreak in the 1850s.
- The locations indicated that were clusters around certain intersections where there were polluted wells. Thus, exposing both the problem and the solution.



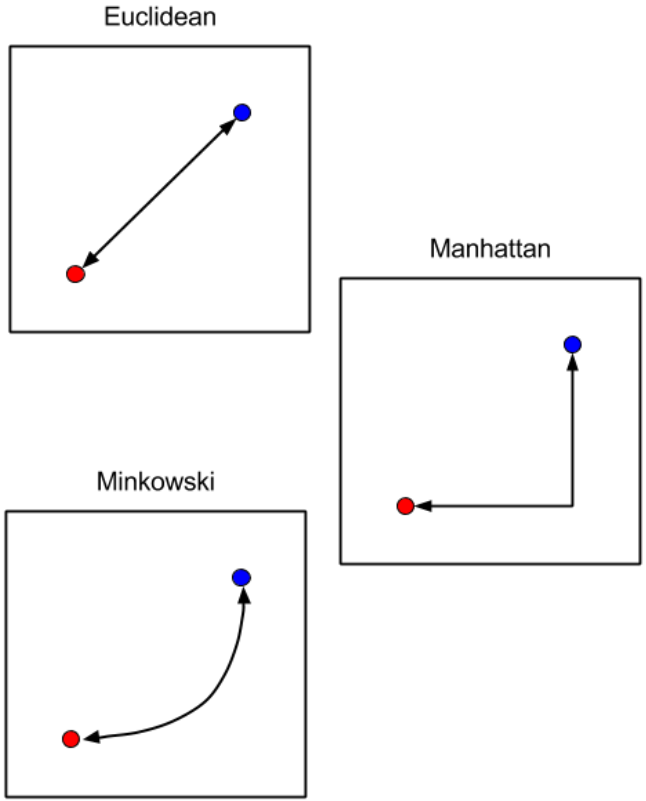
K-MEANS

- One of the most popular clustering algorithms

1. K-means applies to objects that can be represented by points in an n-dimensional space
2. It groups these objects into k classes using an iterative algorithm
3. Each point belong to one and only one cluster
4. The value of k must be given as an input
5. Each cluster is represented by the cluster mean (centroid)

SIMILARITY DISTANCE

- Euclidean distance
  - The most common
- Manhattan distance
  - Approximation to Euclidean distance and cheaper to compute
  - Sum of the absolute differences of their Cartesian coordinates
- Minkowski distance
  - A generalization of both the Euclidean & Manhattan distance



K-MEANS

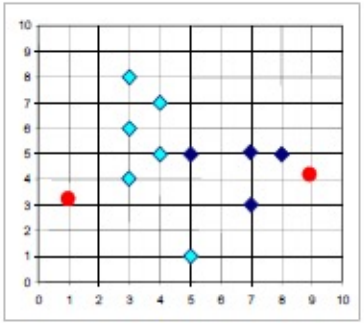
- Default similarity distance: Euclidean distance
- K-means minimizes the total squared distance between each input point (x_i) and its cluster center (c_j)

$$\text{Cost} = \sum_{i=1}^{N} (argmin \, ||xi - cj||^2)$$

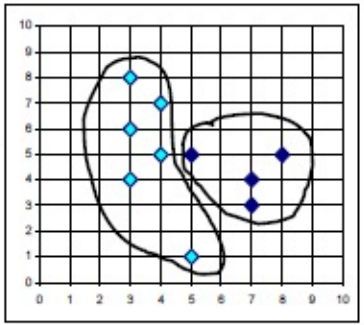
- k-means is an iterative algorithm that first assigns points to clusters and then recomputes the centers of those clusters until finding an optimum solution.

K-MEANS

1. Initialization: arbitrary initialization of the centers

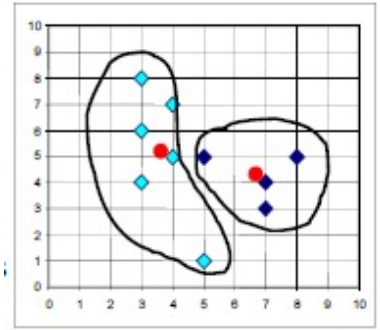


2. Data assignment. Each point is assigned to its closest cluster (center). Ties are broken by randomly assigning the point to one of the clusters. This yields a partitioning of the data.

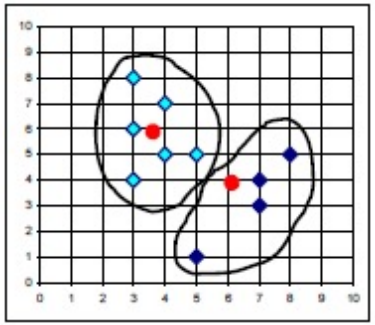


K-MEANS

3. Relocating “centers”. Each cluster representative is moved to the center (arithmetic mean) of the points assigned to it

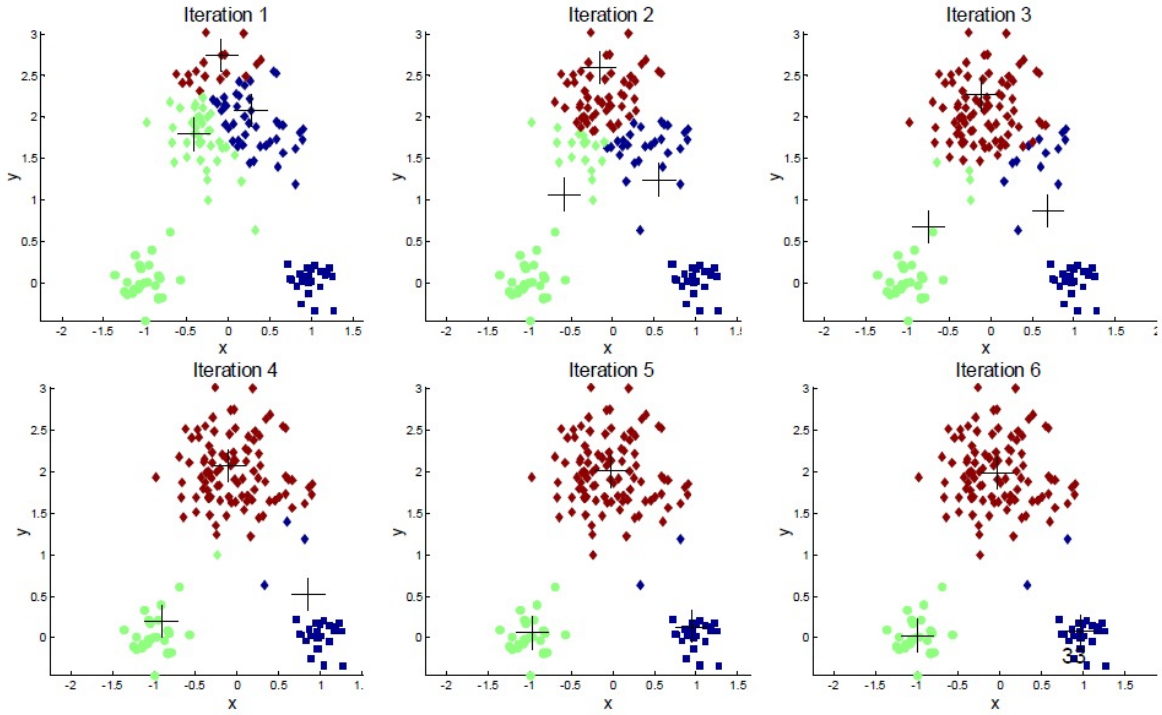


4. Repeat 2 and 3 until the centers do not longer change



*Each iteration requires N*k comparisons. it takes long time for large datasets*

A K-MEANS EXAMPLE OF 3 CLUSTERS



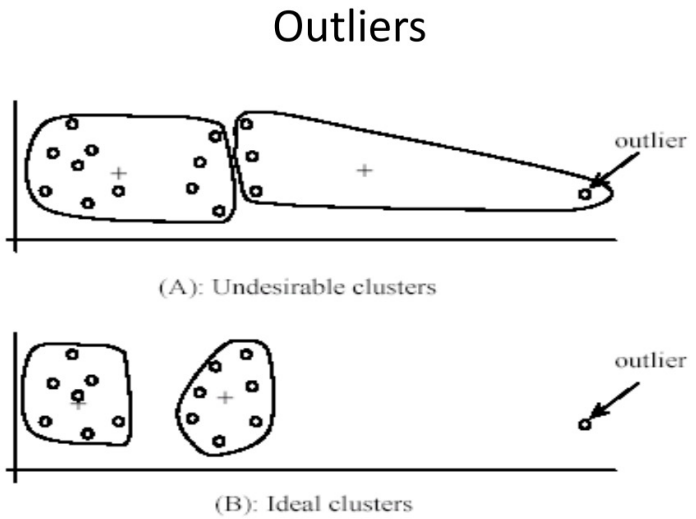
K-MEANS LIMITATIONS

- Sensitive to the initial set-up (location of cluster centers); This might lead to finding local minima.
  - Run multiple times (different initializations) or look for a robust way of initializing the algorithm.

- How to choose k? Hard, unless there is a priori knowledge about the “natural” number of groups present in the data.
  - EDA?
  - K-means starts with one cluster and iteratively increases k until a stopping criterium is met

K-MEANS LIMITATIONS

- Sensitive to outliers. Clusters centers are computed using the “mean” function, which is sensitive to outliers
  - EDA? Outlier removal?



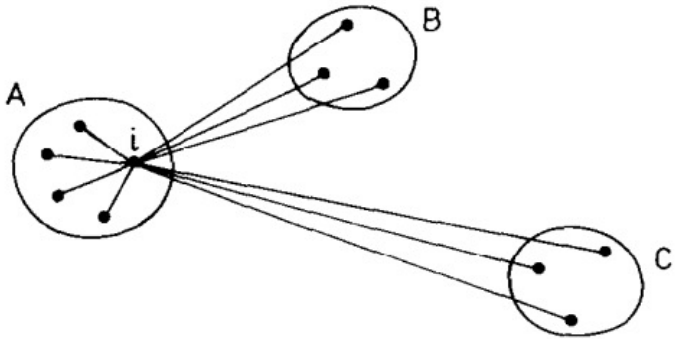
- Risk of empty clusters for high dimensional data; no points close to the chosen center.
  - Dimensionality reduction

- It assumes that the data can be represented by “hyper-spheres”

OPTIMAL K VALUE: SILHOUETTE

- **The average silhouette of the data (Silhouette method)**
  - Silhouette of a data instance→ is a measure of how closely it is matched to data within its cluster and how loosely it is matched to data of the neighbouring cluster
  - A silhouette close to 1 implies the datum is in an appropriate cluster, while a silhouette close to −1 implies the datum is in the wrong cluster.
    - The Silhouette of data instances S(i) is calculated
    - Calculate the average *S* for increasing cluster numbers (e.g., 1 -10)
    - Finally, plot Average *S* – Number of Clusters

OPTIMAL K VALUE: SILHOUETTE



$a(i)$ = average dissimilarity of i to all other objects of A.

$d(i, C)$ = average dissimilarity of i to all objects of C.

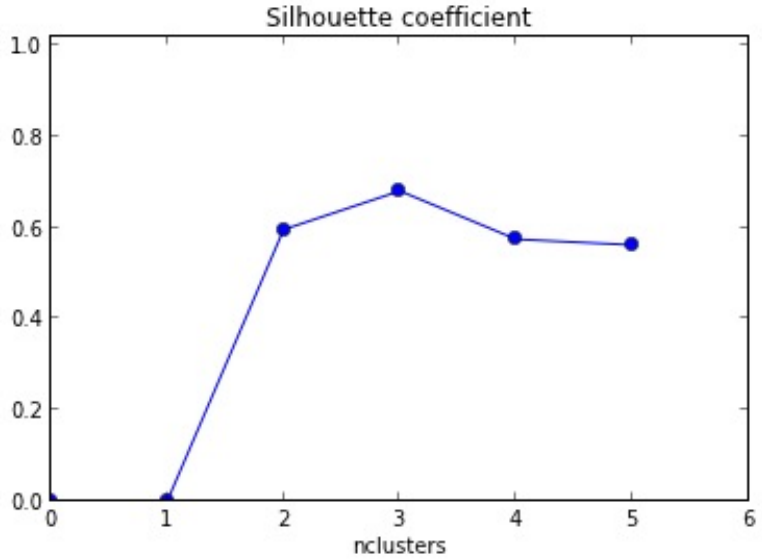
$b(i) = \underset{C \neq A}{\text{minimum}}\ d(i, C)$.

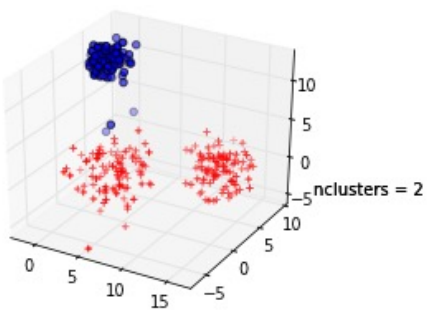
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}.$$

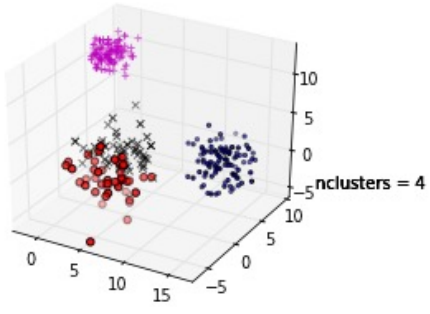
$$s(i) = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i), \\ 0 & \text{if } a(i) = b(i), \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i). \end{cases}$$

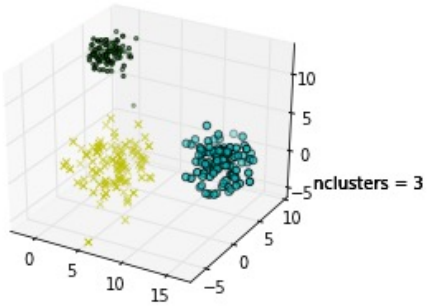
Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20, 53-65.

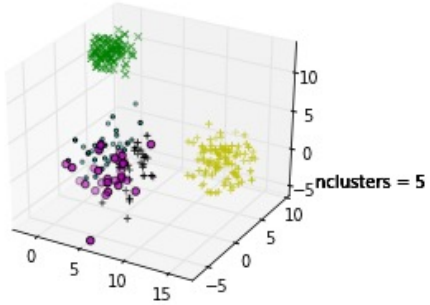
OPTIMAL K VALUE





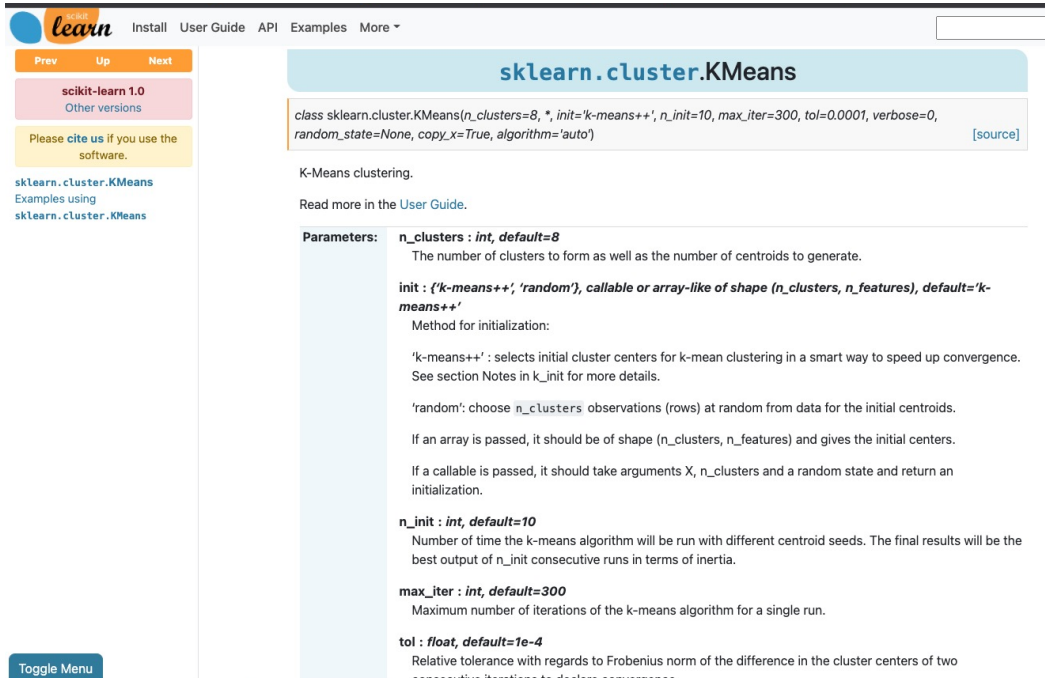






http://stats.stackexchange.com/questions/10540/how-to-interpret-mean-of-silhouette-plot

K-MEANS



Install User Guide API Examples More

Prev Up Next

scikit-learn 1.0
Other versions

Please cite us if you use the software.

sklearn.cluster.KMeans
Examples using
sklearn.cluster.KMeans

Toggle Menu

sklearn.cluster.KMeans

class sklearn.cluster.KMeans(*n_clusters=8*, ***, *init='k-means++'*, *n_init=10*, *max_iter=300*, *tol=0.0001*, *verbose=0*, *random_state=None*, *copy_x=True*, *algorithm='auto'*) [source]

K-Means clustering.

Read more in the User Guide.

Parameters:

n_clusters : *int, default=8*
The number of clusters to form as well as the number of centroids to generate.

init : *{'k-means++', 'random'}, callable or array-like of shape (n_clusters, n_features), default='k-means++'*
Method for initialization:

'k-means++' : selects initial cluster centers for k-mean clustering in a smart way to speed up convergence. See section Notes in k_init for more details.

'random': choose `n_clusters` observations (rows) at random from data for the initial centroids.

If an array is passed, it should be of shape (n_clusters, n_features) and gives the initial centers.

If a callable is passed, it should take arguments X, n_clusters and a random state and return an initialization.

n_init : *int, default=10*
Number of time the k-means algorithm will be run with different centroid seeds. The final results will be the best output of n_init consecutive runs in terms of inertia.

max_iter : *int, default=300*
Maximum number of iterations of the k-means algorithm for a single run.

tol : *float, default=1e-4*
Relative tolerance with regards to Frobenius norm of the difference in the cluster centers of two

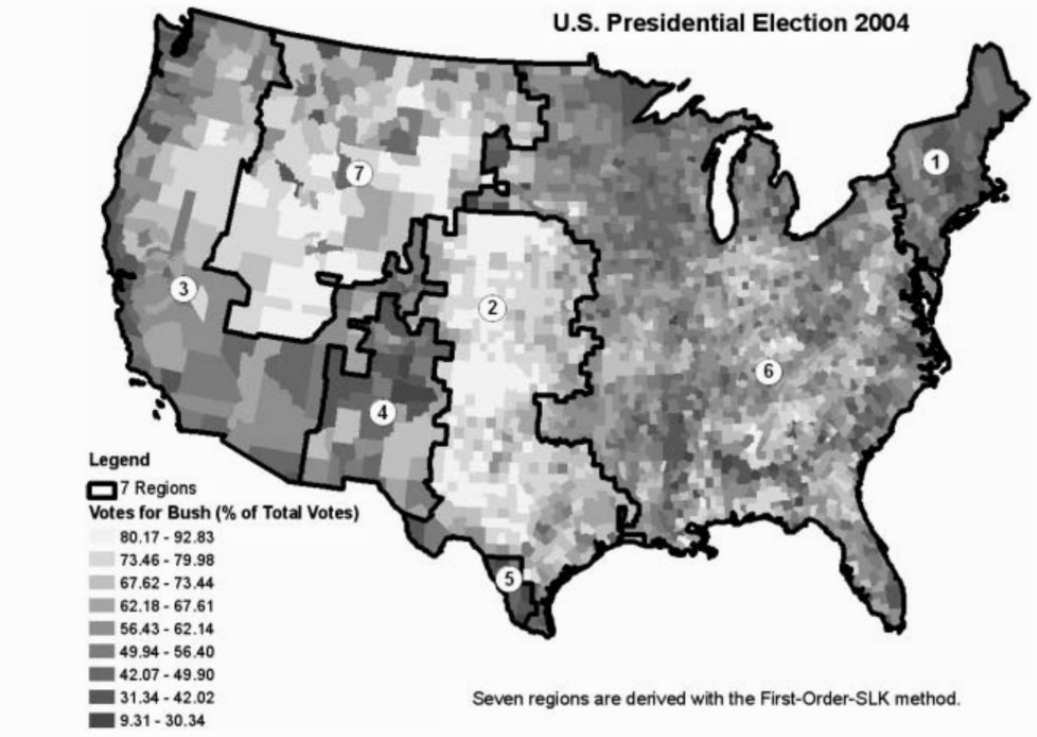
REGIONALIZATION

- Regionalization *is synonymous to spatial classification*
  - organize, visualize, and synthesize the information on multivariate spatial data
  - facilitate the visualization and interpretation of information in maps

- Regionalization *is a zone definition problem (zoning problem)*
  - Technically performed by realizing spatially constrained aggregations

- Spatial contiguity is the critical criterion of a region
  - Has “minor” effects on the loss of detail during aggregations (*due to spatial autocorrelation*)

REGIONALIZATION



Source: Guo, D. (2008). Regionalization with dynamically constrained agglomerative clustering and partitioning (REDCAP). *International Journal of Geographical Information Science*, *22*(7), 801-823.

REGIONALIZATION

- Image Segmentation

Source: Mignotte, M. (2011). A de-texturing and spatially constrained K-means approach for image segmentation. Pattern Recognition Letters, 32(2), 359-367.

MAX-P

- Max-p is also a regionalization algorithm
  - It involves the clustering of spatial units into the **maximum** number of homogeneous regions such that the value of a *spatially extensive regional attribute* is above a predefined threshold value
    - It sets a "*greater than*" condition

- **Resolves 1**: do not know how many regions are needed
- **Resolves 2**: empirically there might be a condition that must be satisfied by every region in order to make them suitable for the analysis
- **Resolves 3**: reduce aggregation bias

MAX-P

- Spatially extensive regional attribute
  - E.g., number of households per region, area per region, population per region, etc.
  - Upon which a minimum threshold value (TH) is imposed.

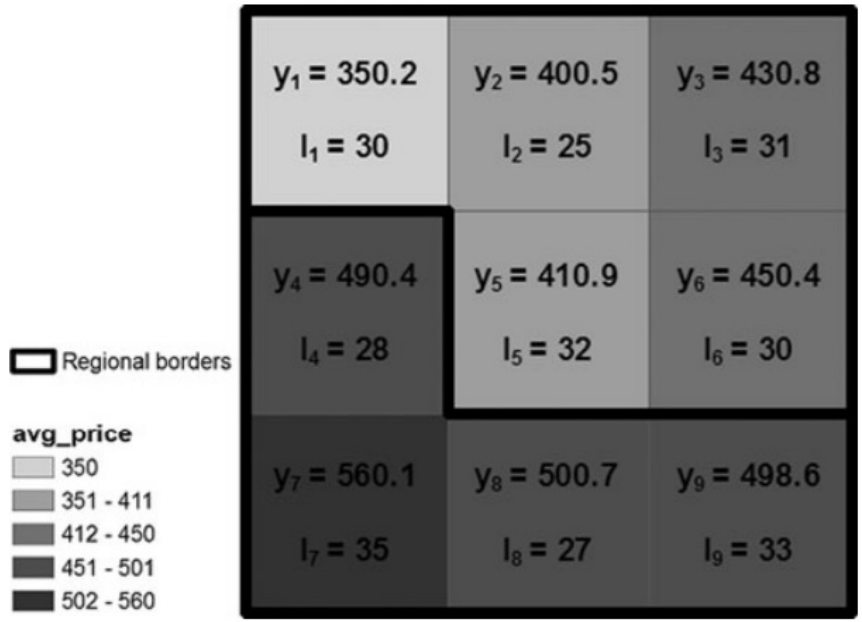


FIGURE 2: Optimal solution for a threshold of 120 houses per region.

Source: Duque, Juan C., Luc Anselin, and Sergio J. Rey. "The max-p-regions problem." *Journal of Regional Science* 52.3 (2012): 397-419.

SPATIO-TEMPORAL CLUSTERING

- Spatial clustering

Date / Station	1992						2011				
	0101	0201	0301	0401				2812	2912	3012	3112
…..	…	…	…	…	…	…	…	…	…	…	…
283 Hupsel	4.9	5.9	6.4	6.4	…	…	…	6.4	5.6	4.1	6.2
286 Nieuw Beerta	6.6	7.9	5.8	5.8	…	…	…	6.8	5.8	4.1	4.8
290 Twente	5.0	6.2	6.2	6.2	…	…	…	6.5	5.6	4.2	5.8
310 Vlissingen	6.1	6.8	6.8	6.8	…	…	…	6.5	7.4	6.0	8.6
319 Westdorpe	5.3	5.9	4.9	4.9	…	…	…	5.9	6.1	5.2	9.7
…..	…	…	…	…	…	…	…	…	…	…	…

SPATIO-TEMPORAL CLUSTERING

- Temporal clustering

Date / Station	1992						2011				
	0101	0201	0301	0401				2812	2912	3012	3112
…..	…	…	…	…	…	…	…	…	…	…	…
283 Hupsel	4.9	5.9	6.4	6.4	…	…	…	6.4	5.6	4.1	6.2
286 Nieuw Beerta	6.6	7.9	5.8	5.8	…	…	…	6.8	5.8	4.1	4.8
290 Twente	5.0	6.2	6.2	6.2	…	…	…	6.5	5.6	4.2	5.8
310 Vlissingen	6.1	6.8	6.8	6.8	…	…	…	6.5	7.4	6.0	8.6
319 Westdorpe	5.3	5.9	4.9	4.9	…	…	…	5.9	6.1	5.2	9.7
…..	…	…	…	…	…	…	…	…	…	…	…

SPATIO-TEMPORAL CLUSTERING

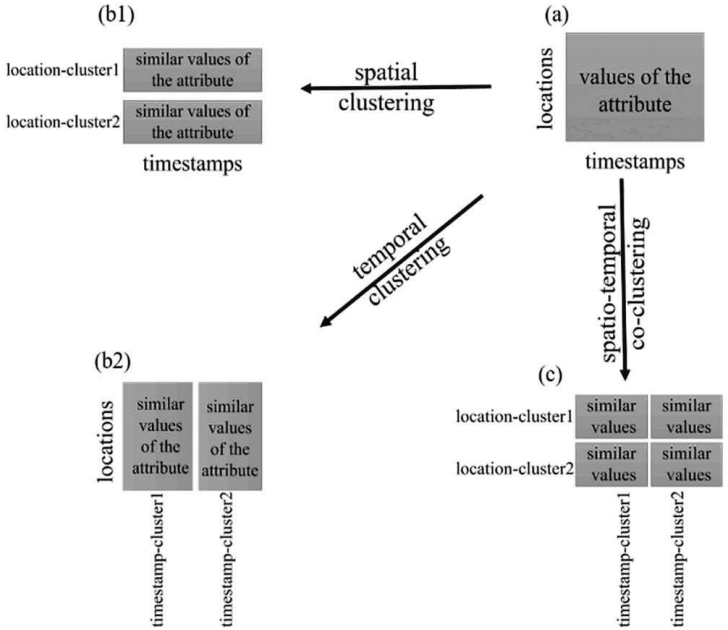
- Clustering from a spatial aspect misses differences in the patterns caused by temporal dynamics and vice versa

Date / Station	1992						2011				
	0101	0201	0301	0401				2812	2912	3012	3112
.....	...	...	...	...	...	...	...	...	...	...	...
283 Hupsel	4.9	5.9	6.4	6.4	...	...	...	6.4	5.6	4.1	6.2
286 Nieuw Beerta	6.6	7.9	5.8	5.8	...	...	...	6.8	5.8	4.1	4.8
290 Twente	5.0	6.2	6.2	6.2	...	...	...	6.5	5.6	4.2	5.8
310 Vlissingen	6.1	6.8	6.8	6.8	...	...	...	6.5	7.4	6.0	8.6
319 Westdorpe	5.3	5.9	4.9	4.9	...	...	...	5.9	6.1	5.2	9.7
.....	...	...	...	...	...	...	...	...	...	...	...

Co-clustering looks for ‘blocks’ (or ‘co-clusters’) of rows and columns that are inter-related (Hartigan, 1972; Benerjee et al., 2007).

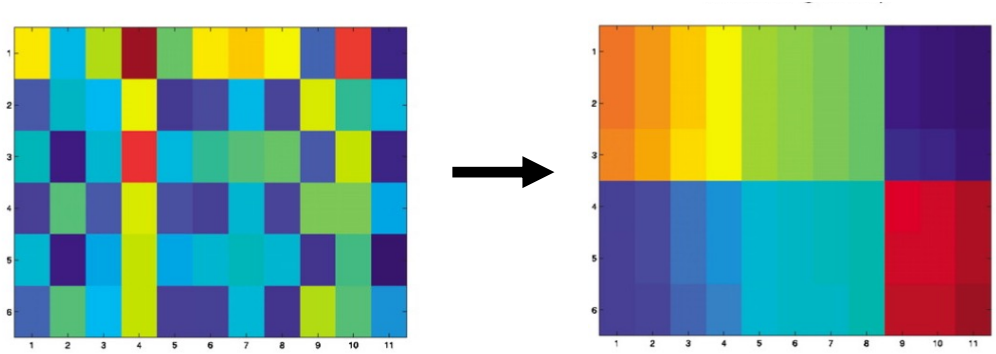
HOW? CO-CLUSTERING

- Bregman block average co-clustering algorithm with I-divergence (BBAC_I)
- Works with any data matrix with positive and real valued elements that represent a joint probability distribution or co-occurrences between two random variables (space and time)



CO-CLUSTERING (2)

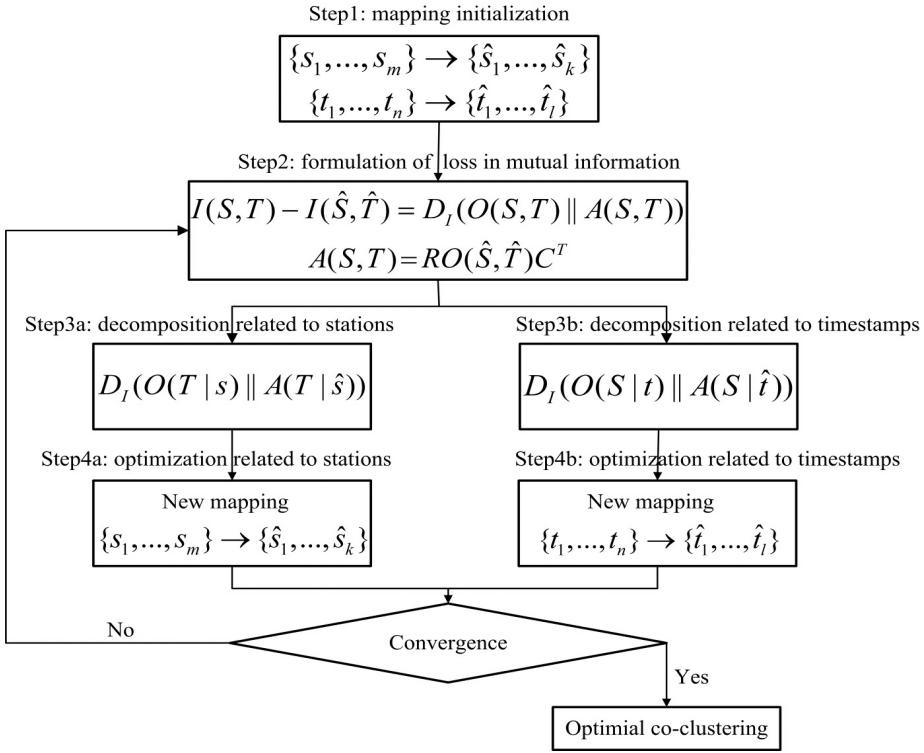
- BBAC_I partitions the data matrix into homogenous blocks by minimizing the loss of information between the original and the co-clustered data matrices



Input data matrix

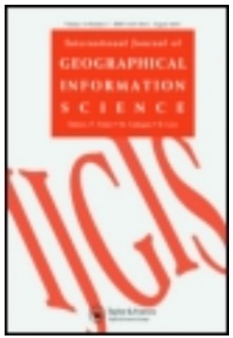
Co-clustered data matrix

CO-CLUSTERING (3)



Inputs:
k and l → number of row and column clusters
Ɛ → threshold loss of mutual information
N → number of replications (local minima)

MORE DETAILS



International Journal of Geographical Information Science

Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tgis20

Co-clustering geo-referenced time series: exploring spatio-temporal patterns in Dutch temperature data

Xiaojing Wu[a], Raul Zurita-Milla[a] & Menno-Jan Kraak[a]

[a] Department of Geo-Information Processing, Faculty of Geo-Information Science and Earth Observation (ITC), University of Twente, Enschede, The Netherlands

Published online: 10 Feb 2015.

JOURNAL OF GEOPHYSICAL RESEARCH
Biogeosciences
AN AGU JOURNAL

JGR

Research Article

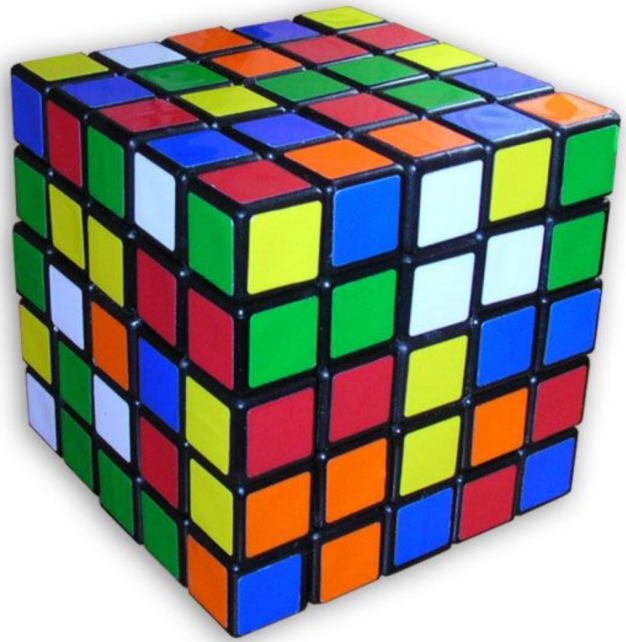
A novel analysis of spring phenological patterns over Europe based on co-clustering

Xiaojing Wu ✉, Raul Zurita-Milla, Menno-Jan Kraak

First published: 9 June 2016 Full publication history

DOI: 10.1002/2015JG003308 View/save citation

TRI-CLUSTERING



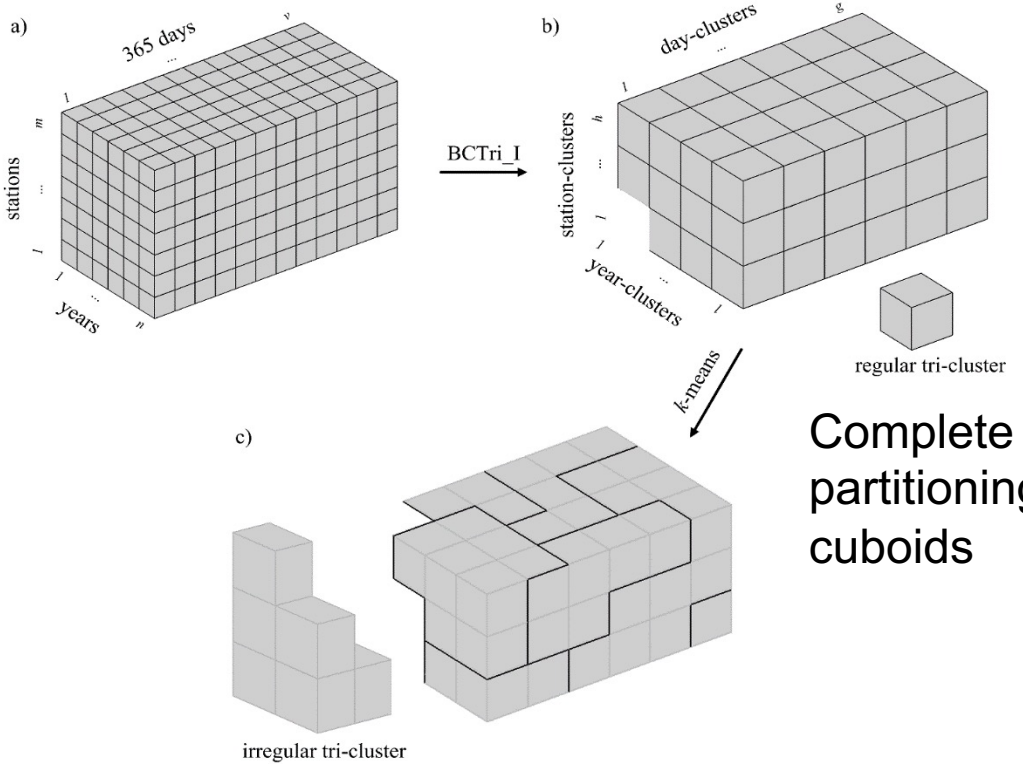
Georeferenced time series (GTS)

GTS with multiple attributes

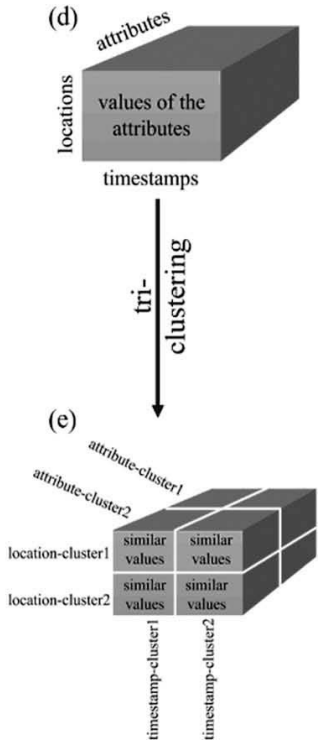
GTS with nested spatial scales (e.g. provinces, states)

GTS with nested temporal scales (e.g. day, year)

TRI-CLUSTERING: BREGMAN CUBOID AVERAGE



Complete partitioning into cuboids



TRI-CLUSTERING (2): ALGORITHM

Algorithm 1 Tri-clustering algorithm

Require: $\mathbf{O} \in \mathbb{R}^{m \times n \times v}$: original data, h: num. of rows clusters, ℓ: num. of columns clusters, g: num. of vector clusters,

Ensure: $\mathbf{R}^* \in \mathbb{R}^{m \times h}$, $\mathbf{C}^* \in \mathbb{R}^{n \times \ell}$ and $\mathbf{T}^* \in \mathbb{R}^{v \times g}$

Random initialization of $\mathbf{R}$, $\mathbf{C}$, $\mathbf{T}$

$\mathbf{T1} \in \mathbb{R}^{nv \times g} \leftarrow$ vertically concatenate of $\mathbf{T}$ n times

while untill the convergence **do**

Updated row clustering:

$\mathbf{O}' \in \mathbb{R}^{m \times nv} \leftarrow$ reshape $\mathbf{O}$

$\mathbf{A} \leftarrow \mathbf{R}(\mathbf{R}^\top \mathbf{O}'\mathbf{T1}/\mathbf{R}^\top \mathbb{1}\mathbf{T1})'\mathbf{T1}^\top$

$D_{I\,i,\cdot} \in \mathbb{R}^h \leftarrow D_I(\mathbf{O}'(i,\cdot)||\mathbf{A}(i,\cdot))$

$\mathbf{R}^* \leftarrow$ binary encoding of $(\arg\min_{j \in [1,h]}\{D_{I\,i,j}\})$

Updated column clustering:

$\mathbf{R1} \in \mathbb{R}^{mv \times h} \leftarrow$ vertically concatenate of $\mathbf{R}^*$ v times

$\mathbf{O}' \in \mathbb{R}^{n \times mv} \leftarrow$ reshape $\mathbf{O}$

$\mathbf{A} \leftarrow \mathbf{C}(\mathbf{C}^\top \mathbf{O}'\mathbf{R1}/\mathbf{C}^\top \mathbb{1}\mathbf{R1})'\mathbf{R1}^\top$

$D_{I\,p,\cdot} \in \mathbb{R}^\ell \leftarrow D_I(\mathbf{O}'(p,\cdot)||\mathbf{A}(p,\cdot))$

$\mathbf{C}^* \leftarrow$ binary encoding of $(\arg\min_{q \in [1,\ell]}\{D_{I\,p,q}\})$

Updated depth clustering:

$\mathbf{C1} \in \mathbb{R}^{mn \times \ell} \leftarrow$ vertically concatenate of $\mathbf{C}^*$ m times

$\mathbf{O}' \in \mathbb{R}^{v \times mn} \leftarrow$ reshape $\mathbf{O}$

$\mathbf{A} \leftarrow \mathbf{T}(\mathbf{T}^\top \mathbf{O}'\mathbf{C1}/\mathbf{T}^\top \mathbb{1}\mathbf{C1})'\mathbf{C1}^\top$

$D_{I\,w,\cdot} \in \mathbb{R}^g \leftarrow D_I(\mathbf{O}'(w,\cdot)||\mathbf{A}(w,\cdot))$

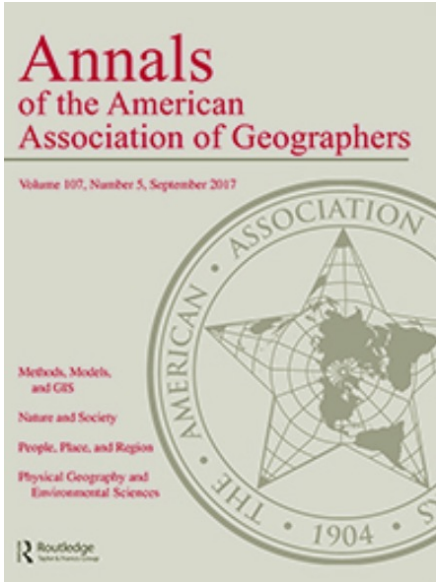
$\mathbf{T}^* \leftarrow$ binary encoding of $(\arg\min_{e \in [1,g]}\{D_{I\,w,e}\})$

$\mathbf{T1} \in \mathbb{R}^{nv \times g} \leftarrow$ vertically concatenate of $\mathbf{T}^*$ n times

end while

Works with any data matrix with positive and real valued elements that represent a joint probability distribution or co-occurrences between three random variables (spatial/temporal)

CO-/TRI-CLUSTERING: MORE DETAILS



Triclustering Georeferenced Time Series for Analyzing Patterns of Intra-Annual Variability in Temperature

Xiaojing Wu, Raul Zurita-Milla, Emma Izquierdo Verdiguier & Menno-Jan Kraak

Pages 1-17 | Received 01 Nov 2016, Accepted 01 Feb 2017, Published online: 22 Jun 2017

CLUSTERING GEO-DATA CUBES (CGC)

https://github.com/esciencecenter-digital-skills/tutorial-cgc