Clustering methods

Donatien Hainaut

IABE data science certificate, March 22

Introduction

- Cluster analysis is part of machine learning, helping to uncover group structures in data.
- Objects are grouped in such a way that the created groups ('clusters') are as much as possible heterogeneous between each-others...
- ...while being homogeneous regarding observations classified within them.
- In actuarial applications, clustering methods can detect dominant sub-populations of policies and the analysis of their claims allows a posteriori to draw a map of insured risks.

Roadmap

- **Principal Component Analysis**
- Factorial component Analysis
- K-means clustering & k-means++
- Clustering with batch k-means*
- Fuzzy clustering
- Spectral clustering

Slides with * : additional material, not seen during the lecture

Basic principle of Factorial methods

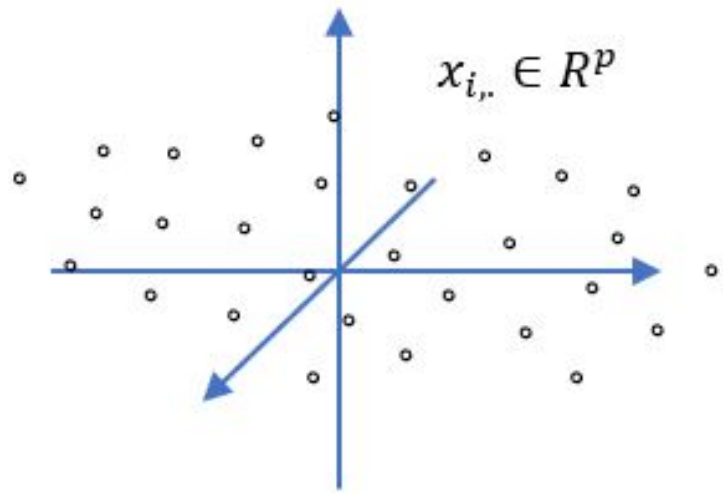
- Objective of factorial methods: to reduce the dimension of the space where the data objects are represented and to visually detect "clusters".
- We consider a a dataset with n numeric objects and p features:

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix}$$

- Main question: is this possible to provide pictures of X in $\mathbb{R}$, $\mathbb{R}^2$ or $\mathbb{R}^3$,...
- There are two ways of looking to X:
  - either row by row
  - either column by column

Basic principle of Factorial methods

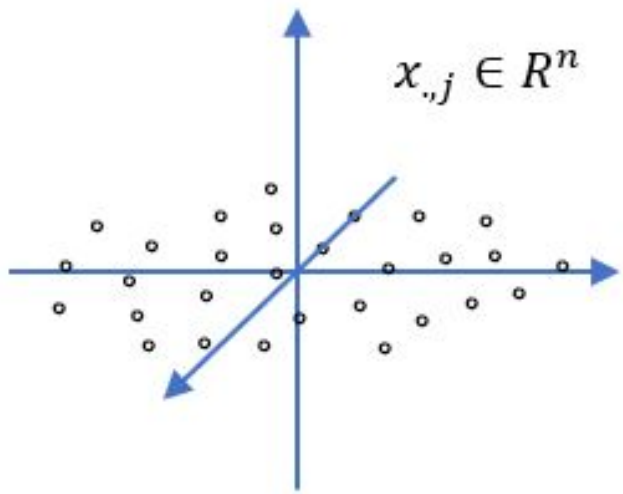
- Row by row analysis in $\mathbb{R}^p$: each row (individual) is a vector $\boldsymbol{x}_{i,.} = (x_{i1}, ..., x_{ip})^\top \in \mathbb{R}^p$



- The aim is to represent individuals in a space of lower dimension to identify clusters of individuals.

Basic principle of Factorial methods

- Column by column analysis in $\mathbb{R}^n$: each column (variable) is a vector $\boldsymbol{x}_{.,j} = (x_{1,j}, ..., x_{n,j})^\top \in \mathbb{R}^n$



- The aim is to represent variables in a space of lower dimension to identify clusters of features.

Basic principle of Factorial methods

Important

Since all further derivations should not depend on translations and scales we work with standardized dataset. If $x^*_{i,j}$ are raw data, we define

$$\mu^*_{.,j} = \frac{1}{n}\sum_{i=1}^{n} x^*_{i,j} \quad \sigma^{*2}_{.,j} = \frac{1}{n}\sum_{i=1}^{n} \left(x^*_{i,j} - \mu_{.,j}\right)^2$$

and standardized $x_{i,j}$ are

$$x_{i,j} = \frac{x^*_{i,j} - \mu^*_{.,j}}{\sqrt{n}\sigma^*_{.,j}} \quad i = 1, ..., n \quad j = 1, ..., p$$

(we divide by $\sqrt{n}$ for mathematical elegance).

Principal Components Analysis

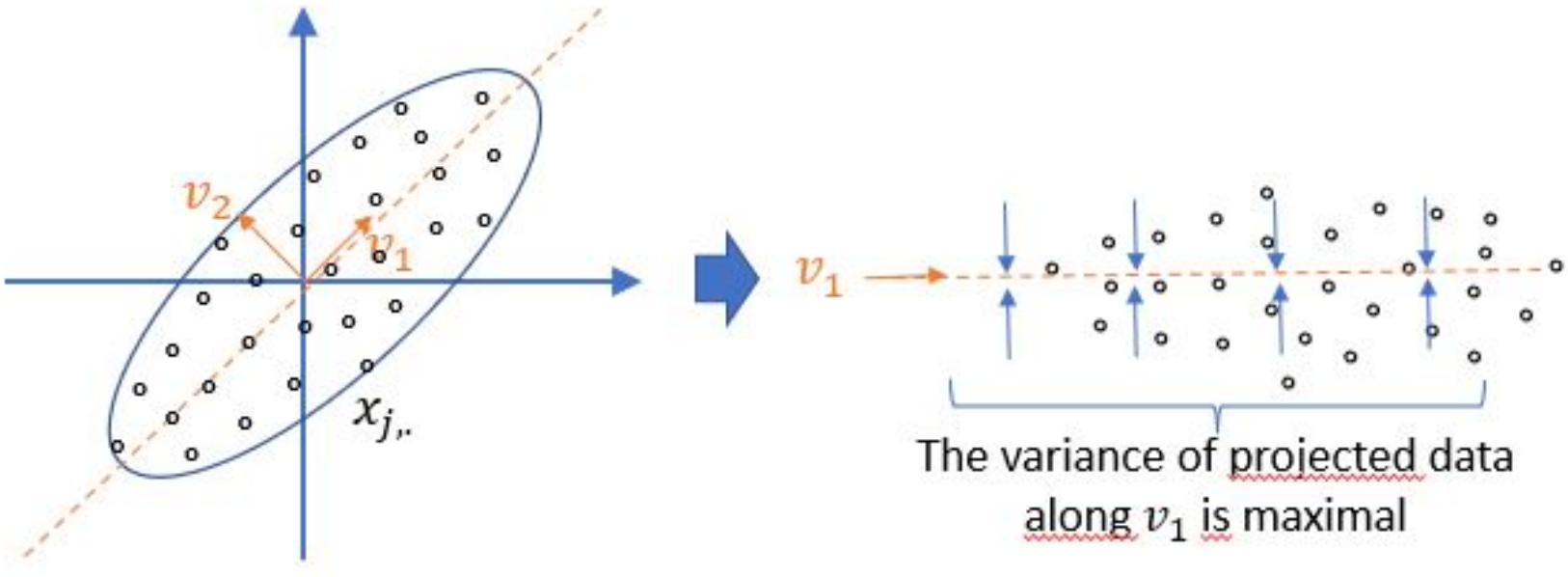
- Row by row analysis in $\mathbb{R}^p$: PCA on rows determines an orthonormal basis $\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \in \mathbb{R}^p$
- $\boldsymbol{v}_1$ explains the direction of the biggest heterogeneity in X, ie the variance of X projected on $\boldsymbol{v}_1$ is maximum

$$\boldsymbol{v}_1 = \arg \max_{||w||_2=1} ||X\boldsymbol{w}||_2^2 = \arg \max_{w^\top w=1} \boldsymbol{w}^\top X^\top X \boldsymbol{w}$$

- $\boldsymbol{v}_k$ the direction of the k^{th} biggest heterogeneity orthogonal to $\boldsymbol{v}_1,...,\boldsymbol{v}_{k-1}$

$$\boldsymbol{v}_k = \arg \max_{||w||_2=1} ||X\boldsymbol{w}||_2^2 \quad s.t. \quad \langle \boldsymbol{w}, \boldsymbol{v}_l \rangle = 0 \quad l = 1, ..., k-1$$

Principal Components Analysis



Principal Components Analysis

PCA on individuals

The principal axis $\boldsymbol{v}_k$ is the normed eigenvector of $X^\top X$ associated with the k^{th} largest eigenvalue, λ_k:

$$\begin{cases} \left(X^\top X\right) \boldsymbol{v}_k = \lambda_k \boldsymbol{v}_k & , \\ \boldsymbol{v}_k^\top \left(X^\top X\right) \boldsymbol{v}_k = \lambda_k & . \end{cases} \tag{1}$$

- The projection of rows on the k^{th} axis is $\boldsymbol{z}_k = X\boldsymbol{v}_k$, the k^{th} principal component.
- λ_k is the inertia ($\propto$variance) of the k^{th} principal component $\boldsymbol{z}_k$. From Eq. (1), it is the sum of square of projections.

Principal Components Analysis*

Sketch of the proof. We consider $k = 1$, $v_1 = \arg \max_{||\boldsymbol{w}||_2=1} ||X\boldsymbol{w}||_2^2$.

If λ_1 is the Lagrange multiplier associated to the constraint $||\boldsymbol{w}||_2 = 1$ then

$$v_1 = \arg \min_{\lambda_1} \max_{\boldsymbol{w}} \underbrace{\boldsymbol{w}^\top X^\top X \boldsymbol{w} - \lambda_1(\boldsymbol{w}^\top \boldsymbol{w} - 1)}_{L}$$

We derive L, the Lagrangian to solve this optimization problem

$$\begin{cases} \frac{\partial L}{\partial \boldsymbol{w}} = 2X^\top X \boldsymbol{w} - 2\lambda_1 \boldsymbol{w} = 0 \\ \frac{\partial L}{\partial \lambda_1} = \boldsymbol{w}^\top \boldsymbol{w} - 1 = 0 \end{cases}$$

and retrieve Equations (1).

Principal Components Analysis

Representation in a subspace of dimension $q \leq p$?

- The projection maximizing the inertia in $\mathbb{R}^q$ is generated by $\boldsymbol{v}_1, ..., \boldsymbol{v}_q$, eigenvectors of $X^\top X$ associated with the q largest eigenvalues $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_q$.
- The quality of the representation is measured by the ratio

$$\tau_q = \frac{\lambda_1 + \lambda_2 + ... + \lambda_q}{\lambda_1 + \lambda_2 + ... + \lambda_p},$$

the percentage of the total inertia, explained by the q first principal axis.

Principal Components Analysis

- Column by Column analysis in $\mathbb{R}^n$: X is a cloud of p points (variables) of $\mathbb{R}^n$. PCA on variables determines a different orthonormal basis $\boldsymbol{u}_1, \boldsymbol{u}_2, ... \in \mathbb{R}^n$.
- $\boldsymbol{u}_k$ the direction of the k^{th} biggest heterogeneity orthogonal to $\boldsymbol{u}_1,...,\boldsymbol{u}_{k-1}$

$$\boldsymbol{u}_k = \arg \max_{||w||_2=1} ||X^\top \boldsymbol{w}||_2^2 \quad s.t. \quad \langle \boldsymbol{w}, \boldsymbol{u}_l \rangle = 0 \quad l = 1, ..., k-1$$

- This is exactly the same problem as the row by row analysis, except that X has to be replaced by $X^\top$.

Principal Components Analysis*

PCA on variables

The principal axis $\boldsymbol{u}_k$ is the normed eigenvector of $XX^\top$ associated with the k^{th} largest eigenvalue, ξ_k:

$$\begin{cases} \left(XX^\top\right)\boldsymbol{u}_k = \xi_k \boldsymbol{u}_k & , \\ \boldsymbol{u}_k^\top \left(XX^\top\right)\boldsymbol{u}_k = \xi_k & . \end{cases} \tag{2}$$

And we have the dual relation

$$\begin{cases} \lambda_k = \xi_k \\ \boldsymbol{v}_k = \frac{1}{\sqrt{\lambda_k}} X^\top \boldsymbol{u}_k \\ \boldsymbol{u}_k = \frac{1}{\sqrt{\lambda_k}} X^\top \boldsymbol{v}_k \end{cases} \tag{3}$$

Principal Components Analysis*

Sketch of the proof Premultipyling by $X^\top$ and X, Eq.'s (2) and (1)

$$\begin{cases} X^\top X \left(X^\top \boldsymbol{u}_k\right) = \xi_k \left(X^\top \boldsymbol{u}_k\right) & , \\ XX^\top \left(X \boldsymbol{v}_k\right) = \lambda_k \left(X \boldsymbol{v}_k\right) & . \end{cases} \tag{4}$$

To each eigenvector $\boldsymbol{v}_k$ of $X^\top X$ (resp. u_k of $XX^\top$) corresponds an eigenvector of $XX^\top$ (resp. $X^\top X$) with the same eigenvalue. Then

$$\begin{cases} \boldsymbol{v}_k = c_k X^\top \boldsymbol{u}_k \\ \boldsymbol{u}_k = d_k X \boldsymbol{v}_k \end{cases}$$

where c_k and d_k are some constants. Since $\boldsymbol{v}_k^\top \boldsymbol{v}_k = \boldsymbol{u}_k^\top \boldsymbol{u}_k = 1$, we have $c_k = d_k = \frac{1}{\sqrt{\lambda_k}}$.

Principal Components Analysis*

Standardization of raw data $x^*_{i,j}$ has some geometrical implications on the analysis in $\mathbb{R}^n$:

- The scalar product of $\boldsymbol{x}_{.,i}$ and $\boldsymbol{x}_{.,j}$ is the correlation $r_{i,j}$ between i^{th} and j^{th} variables

$$\boldsymbol{x}_{.,i}^\top \boldsymbol{x}_{.,j} = \frac{1}{n}\sum_{k=1}^{n} \frac{x^*_{k,i} - \mu^*_{.,i}}{\sigma_{.,i}} \frac{x^*_{k,j} - \mu^*_{.,j}}{\sigma_{.,j}} = r_{i,j}$$

- Norm of $\boldsymbol{x}_{.,j}$ is equal to 1,

$$\|\boldsymbol{x}_{.,j}\|_2^2 = \boldsymbol{x}_{.,j}^\top \boldsymbol{x}_{.,j} = r_{j,j} = 1$$

- **Consequence**: the representation of variables of X in $\mathbb{R}^q$ are all located in a sphere of dimension q and radius equal to 1.

Principal Components Analysis

Example: Churn dataset,
https://www.kaggle.com/sakshigoyal7/credit-card-customers.
Predict that a bank customer is gonna get churned from the credit card service. 10 000 lines.

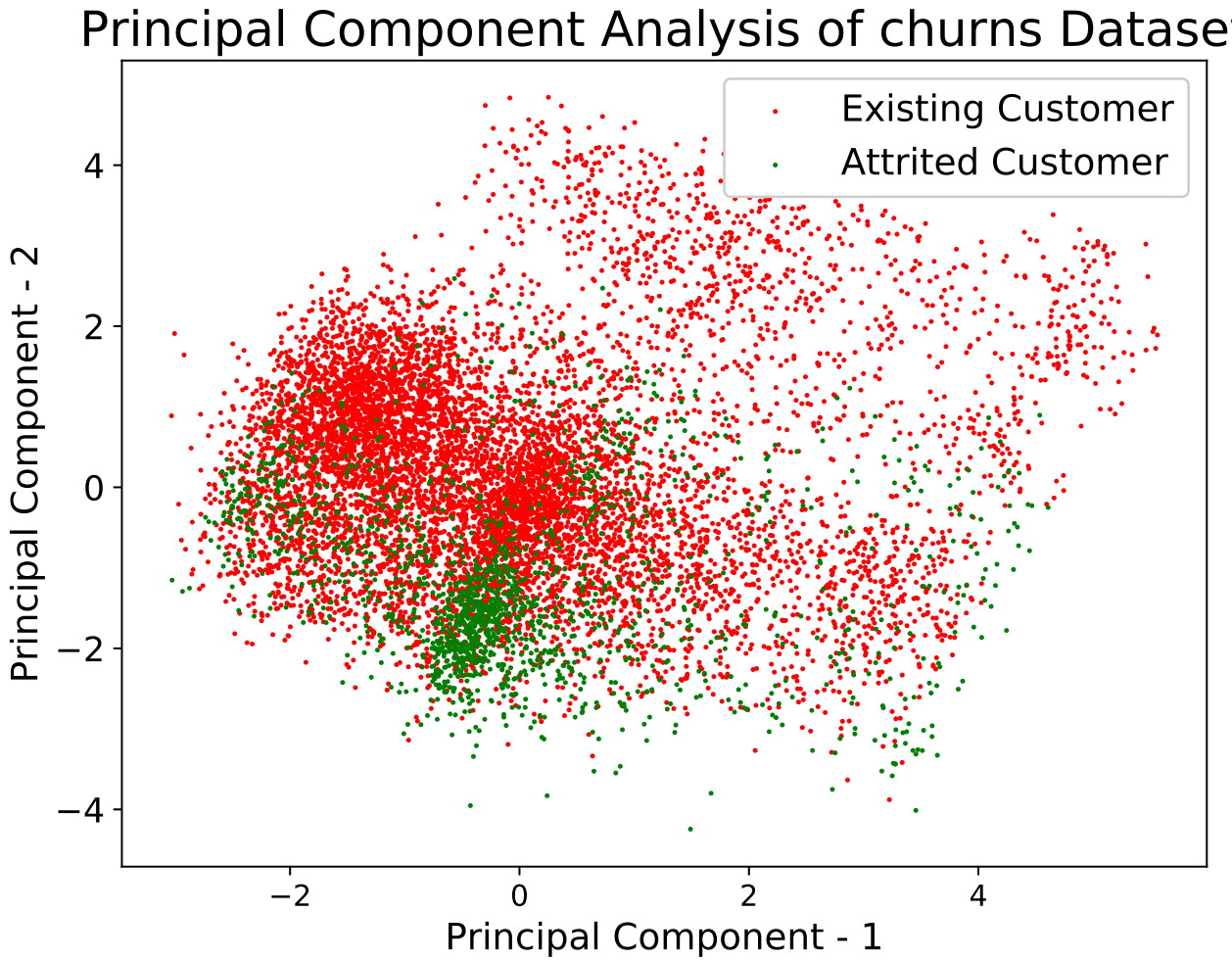
Attrition_Flag	if the account is closed then 1 else 0	Type
Customer_Age	Customer's Age in Years	num
Gender	M=Male, F=Female	cat
Dependent_count	Number of dependents	num
Education_Level	high school, college graduate, etc.	cat
Marital_Status	Married, Single, Divorced, Unknown	cat
Income_Category	Annual Income Category	cat
Card_Category	Blue, Silver, Gold, Platinum	cat
Months_on_book	Period of relationship with bank	num
Total_Relationship_Count	Total no. of products held by the customer	num

Principal Components Analysis

Months_Inactive_12_mon	No. of months inactive in the last 12 months	num
Contacts_Count_12_mon	No. of Contacts in the last 12 months	num
Credit_Limit	Credit Limit on the Credit Card	num
Total_Revolving_Bal	Total Revolving Balance on the Credit Card	num
Avg_Open_To_Buy	Open to Buy Credit Line	num
otal_Amt_Chng_Q4_Q1	Change in Transaction Amount (Q4 over Q1)	num
Total_Trans_Amt	Total Transaction Amount (last 12M)	num
Total_Trans_Ct	Total Transaction Count (Last 12M)	num
Total_Ct_Chng_Q4_Q1	Change in Transaction Count (Q4 over Q1)	num
Avg_Utilization_Ratio	Average Card Utilization Ratio	num

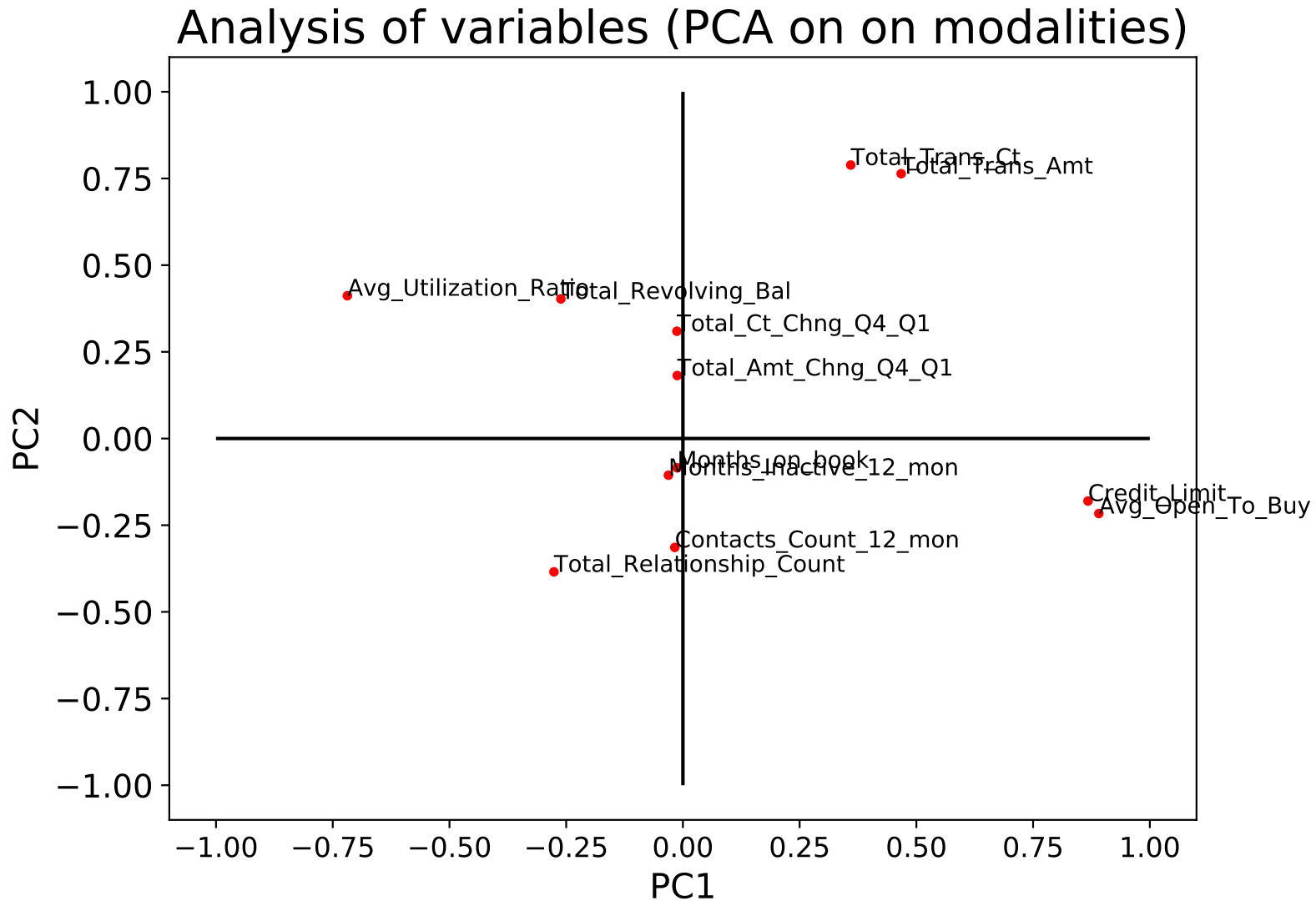
```
from sklearn.preprocessing import StandardScaler
X=StandardScaler().fit_transform(X)
from sklearn.decomposition import PCA
pca_X = PCA(n_components=2.fit_transform(X)
```

Principal Components Analysis



Using PC1 and PC2 as covariates in a logistic regression allows to predict churn with a 84.60% accuracy

Principal Components Analysis



Roadmap

- Principal Component Analysis
- **Factorial component Analysis**
- K-means clustering & k-means++
- Clustering with batch k-means*
- Fuzzy clustering
- Spectral clustering

Factorial Components Analysis of modalities

Most of datasets s are encoded with categorical variables. In order to apply a PCA to detect **clusters of modalities**, we need a new distance.

- The dataset counts n numeric objects with
  - l categorical features,
  - m_k binary modalities for $k = 1, ..., l$.
  - Total number of modalities: $m = \sum_{k=1}^{l} m_k$

Disjunctive table

The information is summarized by a $n \times m$ matrix $D = (d_{i,j})_{i=1...n, j=1...m}$. If the i^{th} policy presents the j^{th} modality then $d_{i,j} = 1$ and $d_{i,j} = 0$ otherwise.

Factorial Components Analysis of modalities

Example: a policy is encoded with two features, $I = 2$,

- by gender (M=male or F=Female), $m_1 = 2$
- by geographic area (U=urban, S=suburban or C=countryside) $m_2 = 3$.

The disjunctive D table is

	Gender		Area		
Policy	M	F	U	S	C
1	1	0	1	0	0
2	0	1	0	0	1
⋮	⋮	⋮	⋮	⋮	⋮

Factorial Components Analysis of modalities

- To study the dependence between the modalities, we calculate the numbers $n_{i,j}$ of individuals sharing modalities i and j, for $i, j = 1, ..., m$.

Burt matrix

The $m \times m$ Burt matrix $B = (n_{i,j})_{i,j=1,...,m}$ is a **contingency** table defined by

$$B = D^\top D\,.$$

- This symmetric matrix is composed of $I \times I$ blocks $B_{k,j}$ for $k, j = 1, ..., I$.
- A block $B_{k,j}$ is the contingency table that crosses the variables k and j.

Factorial Components Analysis of modalities

- **Example (cont'd)**. The Burt matrix is:

		Gender		Area			
		M	F	U	S	C	
Gender	M	$n_{1,1}$	0	$n_{1,3}$	$n_{1,4}$	$n_{1,5}$	$n_{1,.}$
	F	0	$n_{2,2}$	$n_{2,3}$	$n_{2,4}$	$n_{2,5}$	$\vdots$
Area	U	$n_{3,1}$	$n_{3,2}$	$n_{3,3}$	0	0	$\vdots$
	S	$n_{4,1}$	$n_{4,2}$	0	$n_{4,4}$	0	$\vdots$
	C	$n_{5,1}$	$n_{5,2}$	0	0	$n_{5,5}$	$n_{5,.}$
		$n_{.,1}$	$\dots$	$\dots$	$\dots$	$n_{.,5}$	

- $n_{2,2}$: number of women
- $n_{2,3}$: number of women living in an urban environment.
- $n_{1,1} + n_{2,2} = n$ and $n_{3,3} + n_{4,4} + n_{5,5} = n$
- The sum of elements of a block $B_{k,l}$ is equal to number of policies, n.

Factorial Components Analysis of modalities

χ^2-distance

The dependence between rows i and i' is measured by

$$\chi^2(i,i') = \sum_{j=1}^{m} \frac{n}{n_{.,j}} \left(\frac{n_{i,j}}{n_{i,.}} - \frac{n_{i',j}}{n_{i',.}} \right)^2 \quad i,i' \in \{1,...,m\}.$$

- As we prefer to work with Euclidian distances, we calculate the weighted Burt matrix B' of values $b_{i,j}$:

$$\begin{aligned} b_{i,j} &:= \frac{n_{i,j}}{\sqrt{n_{i,.}\, n_{.,j}}} \\ &= \frac{n_{i,j}}{l\sqrt{n_{i,i}\, n_{j,j}}} \end{aligned}$$

as $n_{i,.} = l\, n_{i,i}$ and $n_{.,j} = l\, n_{j,j}$.

Factorial Components Analysis of modalities

- If C is the matrix $C = diag\left(n_{11}^{-\frac{1}{2}}..n_{mm}^{-\frac{1}{2}}\right)$ then $B' = \frac{1}{I}CBC$.
- If $\boldsymbol{b}_i = (b_{i,1}, ..., b_{i,2})^\top$ is the i^{th} line of B', the χ^2 distance between rows (i, i') become:

$$\begin{aligned}\chi^2(i, i') &= \sum_{j=1}^{m} (b_{i,j} - b_{i',j})^2 \\ &= \left\| \boldsymbol{b}_i - \boldsymbol{b}_{i'} \right\|_2^2\end{aligned}$$

- As each modality of the dataset is encoded by a vector $(\boldsymbol{b}_i)_{i=1,...,m} \in \mathbb{R}^m$, we can represent the m modalities in a space of lower dimension using a Principal Component Analysis.

Factorial Components Analysis of modalities

Summary

- Calculate the **disjunctive table** D, (If the i^{th} policy presents the j^{th} modality :$d_{i,j} = 1$ else $d_{i,j} = 0$).
- Calculate the **Burt matrix**

$$B = D^{\top} D\,.$$

- **Weight the Burt matrix**. If $C = diag\left(n_{11}^{-\frac{1}{2}}..n_{mm}^{-\frac{1}{2}}\right)$ then

$$B' = \frac{1}{I} CBC.$$

- **Perform a PCA** on lines of B' to represent the m modalities in a space of lower dimension.

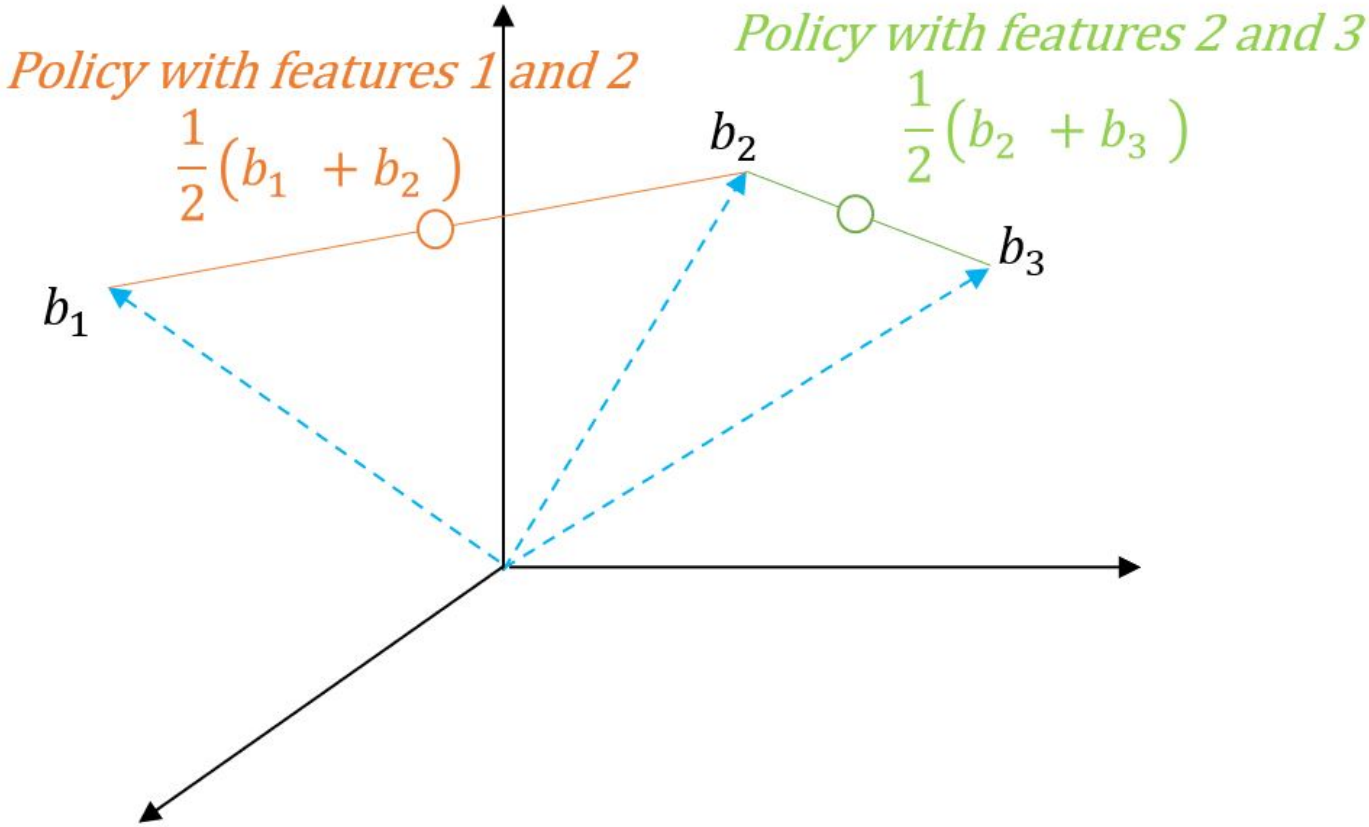
Factorial Components Analysis of individuals

- We propose a novel approach based on the weighted Burt matrix to analyze individuals
- The k^{th} modality corresponds to a vector $\boldsymbol{b}_k$ $\mathbb{R}^m$, the k^{th} line of B'.
- The i^{th} contract with multiple modalities can then be identified by the center of gravity, $\boldsymbol{x}_i$, of points with coordinates stored in the corresponding lines of B':

$$\boldsymbol{x}_i = D_{i,.} B' \,/\, I \quad \text{for } i = 1, ..., n$$

- If X is a matrix with lines $\boldsymbol{x}_i$, we can apply the PCA on X.

Factorial Components Analysis of individuals

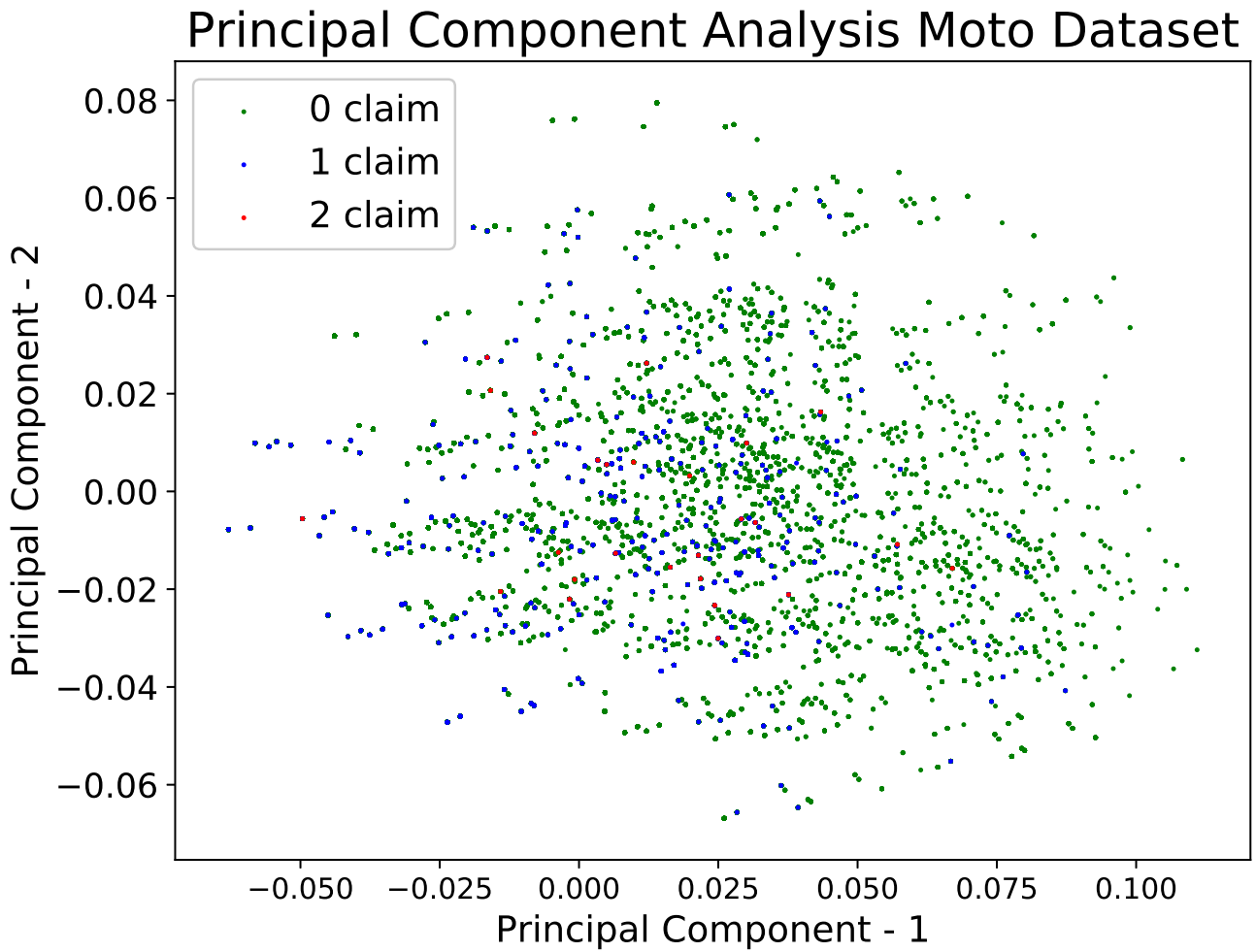


Factorial Components Analysis of individuals

Example: Motorcycles insurances dataset (Wasa 94-98)

Driver's age		9 categories
Age of vehicle		4 categories
Gender		M: Male K: Female
Geographic area	1	Central and semi-central parts of Sweden's three largest cities
	2	Suburbs plus middle-sized cities
	3	Lesser towns, except those in 5 or 7
	4	small towns and countryside
	5	Northen towns
	6	Northen countryside
	7	Gotland (Sweden's largest island)
Vehicle class	1-7	Rating based on KWx100/Kg
Claim counts		integer
Duration		Float, exposure

Factorial Components Analysis of individuals



Deviance GLM with 24 initial variables: **AIC, 7161**. Deviance with 21 PC's : **AIC, 7155**.

Roadmap

- Principal Component Analysis
- Factorial component Analysis
- **K-means clustering & k-means++**
- Clustering with batch k-means*
- Fuzzy clustering
- Spectral clustering

The k-means algorithm

- PCA and FCA allows to vizualize dataset in a low dimension space and to visually detect clusters of individuals or modalities.
- However, PCA and FCA do no provide a systematic method for creating clusters.

Algorithms like the K-means, Fuzzy or spectral clustering offer a systematic approach for identifying potential clusters.

The k-means algorithm

- We again consider a dataset with n numeric objects and p features stored in a matrix

$$X = \begin{pmatrix} x_{1,1} & \cdots & x_{1,p} \\ \vdots & \ddots & \vdots \\ x_{n,1} & \cdots & x_{n,p} \end{pmatrix}$$

- Lines of this matrix are in this section denoted by $(\boldsymbol{x}_i)_{i=1,\ldots,n}$ and correspond to individuals/objects.
- The k-means algorithm is based on the concept of centroids (center of gravity of a cluster of objects).

The k-means algorithm

- The coordinates of the u^{th} centroid is contained in a vector $\boldsymbol{c}_u = (c_1^u, \ldots, c_p^u)$ for $u = 1, ..., k$.
- For a distance $d(.,.)$ and k centroids, the clusters S_u for $u = 1, ..., k$ contains closest objects to $\boldsymbol{c}_u$

$$S_u = \{\boldsymbol{x}_i \ : \ d(\boldsymbol{x}_i, \boldsymbol{c}_u) \leq d(\boldsymbol{x}_i, \boldsymbol{c}_j) \ \forall j \in \{1, ..., k\}\} \qquad (5)$$

- The center of gravity of S_u is a p vector $\boldsymbol{g}_u = \left(g_1^u, \ldots, g_p^u\right)$ such that

$$\boldsymbol{g}_u = \frac{1}{|S_u|} \sum_{\boldsymbol{x}_i \in S_u} \boldsymbol{x}_i \, .$$

The k-means algorithm

The center of gravity of the full dataset is $\boldsymbol{g} = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{x}_i$. the global inertia is

$$I_X = \frac{1}{n}\sum_{i=1}^{n} d\left(\boldsymbol{x}_i, \boldsymbol{g}\right)^2 \ ,$$

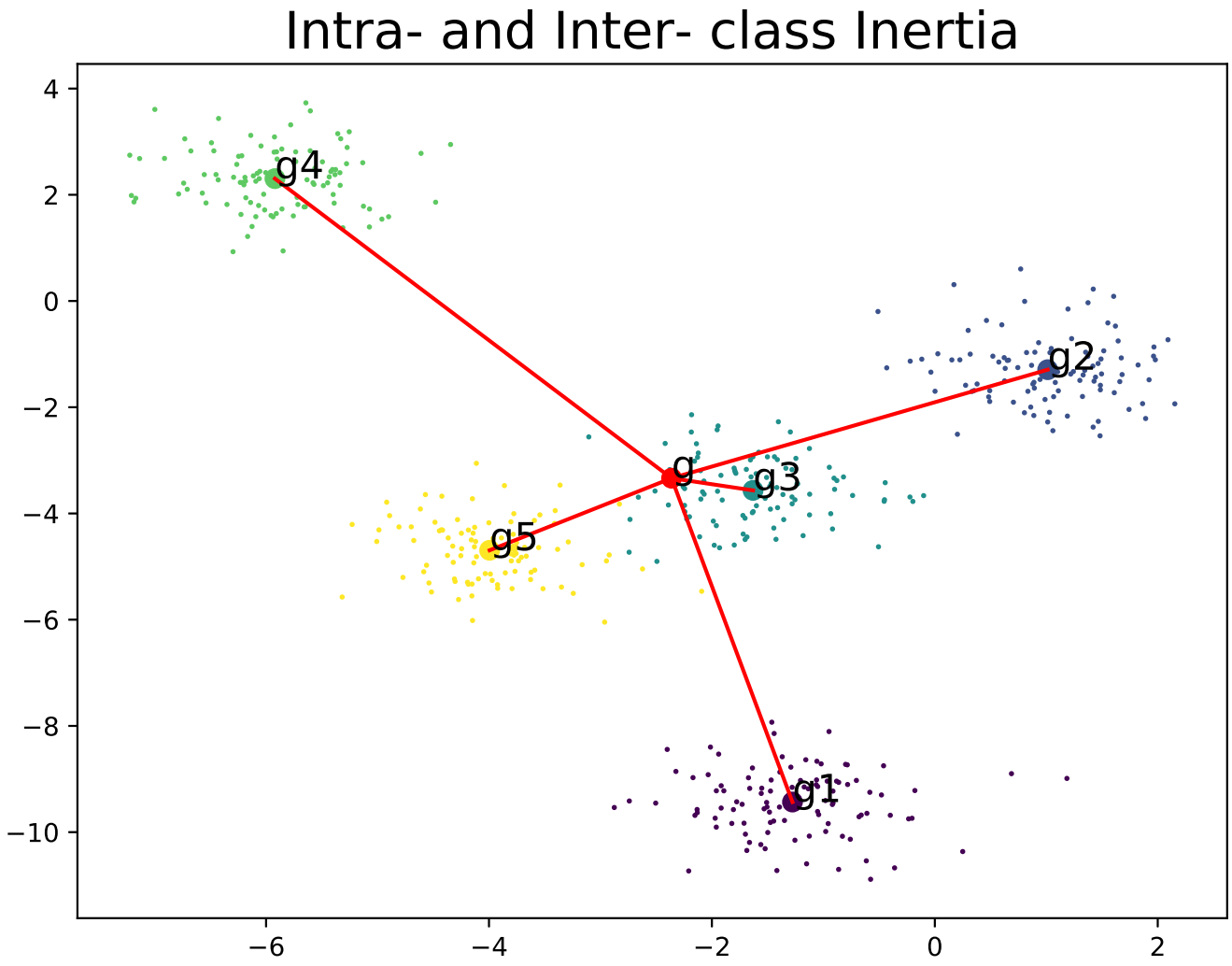
The interclass inertia I_c is the inertia of the cloud of centers of gravity:

$$I_c \ = \ \sum_{u=1}^{k} \frac{|S_u|}{n} d\left(\boldsymbol{g}_u, \boldsymbol{g}\right)^2 \ ,$$

whereas the intraclass inertia I_a is the sum of clusters inertiae:

$$I_a \ = \ \frac{1}{n}\sum_{u=1}^{k}\sum_{\boldsymbol{x}_i \in S_u} d\left(\boldsymbol{x}_i, \boldsymbol{g}_u\right)^2 \ .$$

The k-means algorithm



The k-means algorithm

- According to the König-Huyghens theorem : $I_X = I_c + I_a$.
- An usual criterion of classification consists to seek for a partition of X minimizing the intraclass inertia I_a to form homogeneous clusters on average.
- This is equivalent to determine the partition maximizing the interclass inertia, I_c

Warning

Finding the partition that minimizes the intraclass inertia is computationally difficult (NP-hard).

Solution: heuristic procedures converging quickly to a local optimum like k-means.

The k-means algorithm

Initialization:

Randomly set up initial positions of centroids $\boldsymbol{c}_1(0),...,\boldsymbol{c}_k(0)$.

Main procedure:

For $e = 0$ to maximum epoch, e_{max}

Assignment step:

For $i = 1$ to n

1) Assign $\boldsymbol{x}_i$ to the closest cluster $S_u(e)$, $u \in \{1, ..., k\}$

End loop on data set, i.

Update step:

For $u = 1$ to k

2) Calculate the new centroids $\boldsymbol{c}_u(e+1)$ of $S_u(e)$

$$\boldsymbol{c}_u(e+1) = \frac{1}{|S_u(e)|} \sum_{\boldsymbol{x}_i \in S_u(e)} \boldsymbol{x}_i \, .$$

End loop on centroids, u.

End loop on epochs e

The k-means algorithm

K-means

The k-means algorithm proceeds by alternating between two steps.

- In the assignment step of the e^{th} iteration, we associate $\boldsymbol{x}_i$ to a cluster $S_u(e)$ whose centroid $\boldsymbol{c}_u(e)$ is the nearest.
- Update step, we calculate the new means $\boldsymbol{g}_u(e)$ to be the centroids $\boldsymbol{c}_u(e+1)$ of observations in new clusters

- At each iteration, we can prove that the intraclass inertia is reduced.
- The algorithm converges when the assignments no longer change.

The k-means++ algorithm

- There is no guarantee that a global optimum is found using this algorithm.
- The k-means++ algorithm of Arthur and Vassilvitskii (2007) uses an heuristic to find centroid seeds for k-means clustering.
- The idea consists to initialize the k-means heuristic by selecting centroids that are well scattered.
- Centroids are chosen iteratively and randomly among the dataset with a probability proportional to the distance to the last initialized centroid.

The k-means++ algorithm

Initialization :

Select an observation uniformly at random from the data set, X.

The chosen observation is the first centroid, $\boldsymbol{c}_1(0)$.

Main procedure:

For $j = 2$ to k

For $i = 1$ to n

1) Calculate the distance $d(\boldsymbol{x}_i, \boldsymbol{c}_{j-1}(0))$ from $\boldsymbol{x}_i$ to $\boldsymbol{c}_{j-1}(0)$.

End loop on dataset, i

2) Select the next centroid, $\boldsymbol{c}_j(0)$ at random from X with probability

$$\frac{d^2(\boldsymbol{x}_i, \boldsymbol{c}_{j-1}(0))}{\sum_{i=1}^{n} d^2(\boldsymbol{x}_i, \boldsymbol{c}_{j-1}(0))} \quad i = 1, ..., n.$$

End loop on k

K-means clustering

Example: motorcycle data set.

- As the dataset is made up categorical variables, the matrix X is the matrix of coordinates of policies, $\boldsymbol{x}_i$, in the weighted Burt space:

$$\boldsymbol{x}_i = D_{i,.} B' / l \quad \text{for } i = 1, ..., n$$

- We run the K-means clustering (20 runs). With only 15 clusters, we achieve a Deviance of 5888 (GLM : 5762)

```
from sklearn.cluster import KMeans
n_clus = 15
km = KMeans(init="random" or ‘‘kmeans++’’,
n_clusters=n_clus, n_init=20,max_iter  = 300)
```

K-means clustering

Cluster	Gender	Zone	Class	Owners Age	Vehicule Age	Frequency
0	M	2.0	5.0	40 - 49	11 - 20	0.0164
1	M	4.0	5.0	50 - 59	11 - 20	0.0082
2	M	3.0	3.0	40 - 49	5 - 10	0.0098
3	K	4.0	3.0	40 - 49	5 - 10	0.0109
4	M	3.0	6.0	20 - 29	5 - 10	0.0452
5	M	3.0	1.0	40 - 49	21 -	0.0057
6	M	4.0	4.0	50 - 59	11 - 20	0.004
7	M	4.0	1.0	40 - 49	21 -	0.0036
8	M	3.0	4.0	20 - 29	0 - 4	0.0377
9	M	4.0	3.0	40 - 49	5 - 10	0.0078
10	K	4.0	3.0	40 - 49	0 - 4	0.0118
11	K	4.0	3.0	40 - 49	11 - 20	0.006
12	M	4.0	3.0	40 - 49	0 - 4	0.0091
13	M	4.0	5.0	40 - 49	11 - 20	0.0027
14	M	4.0	3.0	50 - 59	11 - 20	0.0041

Roadmap

- Principal Component Analysis
- Factorial component Analysis
- K-means clustering & k-means++
- **Clustering with batch k-means***
- Fuzzy clustering
- Spectral clustering

Mini-batch k-means*

- For large datasets, the computation time of k-means increases because of its constraint of needing the whole dataset in main memory.
- Mini Batch k-means algorithm's main idea is to use small random batches of data with a fixed size, so they can be stored in memory.
- Each iteration a new random sample from the dataset is obtained and used to update the clusters...
- ...taking care of deprecating previous coordinates according to a learning speed.

Mini-batch k-means*

Initialization:

Randomly set up initial positions of k centroids

Initialize clusters $S_1 = ... = S_k = \emptyset$

Main procedure:

For $e = 0$ to maximum epoch, e_{max}

Random sampling of the batch dataset M of size b

Initialize sample clusters $S_1^{new} = ... = S_k^{new} = \emptyset$

Assignment step:

For $i = 1$ to b

1) Assign i^{th} policy to cluster S_u^{new} where

$$S_u^{new} = \{u \ : \ d(\boldsymbol{x}_i \, , \boldsymbol{c}_u(e)) \leq d(\boldsymbol{x}_i \, , \boldsymbol{c}_j(e)) \ \forall j \in \{1, ..., k\}\} \, .$$

End loop on batch dataset, i.

Mini-batch k-means*

Update step:

For $u = 1$ to k

2) Calculate the centroids of the batch assigned to S_u^{new}:

$$\boldsymbol{c}_u^{new} = \frac{1}{|S_u^{new}|} \sum_{i \in S_u^{new}} \boldsymbol{x}_i \,,$$

3) Let $\eta_u(e) = \frac{|S_u^{new}|}{|S_u| + |S_u^{new}|}$. Centroids $\boldsymbol{c}_u(e+1)$ of S_u are:

$$\boldsymbol{c}_u(e+1) = (1 - \eta_u(e))\, \boldsymbol{c}_u(e) + \eta_u(e) \boldsymbol{c}_u^{new} \,,$$

4) Merge S_u and S_u^{new}

End loop on centroids, u.

End loop on epochs e

Mini-batch k-means*

Example: motorcycle data set.

- As the dataset is made up categorical variables, the matrix X is the matrix of coordinates of policies, $\boldsymbol{x}_i$, in the weighted Burt space:

$$\boldsymbol{x}_i = D_{i,.} B' / l \quad \text{for } i = 1, ..., n$$

- We run the batch K-means clustering (20 runs). With only 15 clusters, we achieve a Deviance of 5854 (GLM : 5762)

```
from sklearn.cluster import  MiniBatchKMeans
n_clus = 15
km = MiniBatchKMeans(init="random",n_clusters=n_clus,
                n_init=20,max_iter  = 300,batch_size=5000)
```

Mini-batch k-means*

Cluster	Gender	Zone	Class	Owners Age	Vehicule Age	Frequency
0	M	1.0	4.0	20 - 29	0 - 4	0.0551
1	M	4.0	3.0	50 - 59	5 - 10	0.0099
2	K	4.0	3.0	40 - 49	11 - 20	0.006
3	M	3.0	6.0	40 - 49	11 - 20	0.0104
4	M	4.0	6.0	20 - 29	5 - 10	0.0383
5	M	3.0	3.0	40 - 49	11 - 20	0.0043
6	M	1.0	4.0	40 - 49	5 - 10	0.0141
7	M	4.0	1.0	40 - 49	21 -	0.0034
8	M	4.0	4.0	50 - 59	11 - 20	0.0032
9	M	4.0	3.0	50 - 59	0 - 4	0.0109
10	M	4.0	2.0	60 - 69	11 - 20	0.0064
11	M	4.0	6.0	40 - 49	11 - 20	0.0048
12	K	4.0	3.0	40 - 49	0 - 4	0.0112
13	M	2.0	3.0	40 - 49	0 - 4	0.0091
14	M	4.0	5.0	40 - 49	11 - 20	0.0066

Roadmap

- Principal Component Analysis
- Factorial component Analysis
- K-means clustering & k-means++
- Clustering with batch k-means*
- **Fuzzy clustering**
- Spectral clustering

Fuzzy clustering

- In non-fuzzy clustering (also known as hard clustering), data is divided into distinct clusters, where each data point can only belong to exactly one cluster.
- In fuzzy clustering, data points can potentially belong to multiple clusters.
- The fuzzy k-means algorithm attempts to partition a finite collection of n elements into a collection of k fuzzy clusters, S_u for $u = 1, ..., k$.
- The algorithm returns a list of k cluster centres $\{\boldsymbol{c}_1, ..., \boldsymbol{c}_k\}$ and a partition matrix W of "membership" $w_{i,j}$ for $i = 1, ..., n$ and $j = 1, ..., k$. The $w_{i,j}$ tells the degree to which the i^{th} policy belongs to cluster S_j.

Fuzzy clustering

Fuzzy clustering

The fuzzy k-means aims to minimize an objective function:

$$\arg\min \sum_{i=1}^{n} \sum_{j=1}^{k} (w_{i,j})^m \, d(\boldsymbol{x}_i, \boldsymbol{c}_j)$$

where

$$w_{i,j} = \frac{1}{\sum_{u=1}^{k} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{c}_j)}{d(\boldsymbol{x}_i, \boldsymbol{c}_u)} \right)^{\frac{2}{m-1}}} \in [0, 1].$$

The hyper-parameter $m \in \mathbb{R}^+$ with $m \geq 1$ is called the fuzzifier.

The fuzzifier, m, determines the level of cluster fuzziness. A large (resp. small) m results in small (resp. high) membership values $w_{i,j}$.

Fuzzy clustering

Initialization:

Randomly set up initial positions of centroids $\boldsymbol{c}_1(0),\ldots,\boldsymbol{c}_k(0)$.

Main procedure:

For $e = 0$ to maximum epoch, e_{max}

Assignment step:

For $i = 1$ to n

1) Calculate membership of the i^{th} policy to $S_j(e)$

$$w_{i,j} = \left(\sum_{u=1}^{k} \left(\frac{d(\boldsymbol{x}_i, \boldsymbol{c}_j)}{d(\boldsymbol{x}_i, \boldsymbol{c}_u)} \right)^{\frac{2}{m-1}} \right)^{-1}.$$

End loop on data set, i.

Fuzzy clustering

Update step:
For $u = 1$ to k
2) Update centroids $\boldsymbol{c}_u(e+1)$ of $S_u(e)$:

$$\boldsymbol{c}_u(e+1) = \frac{\sum_{i=1}^{n} w_{i,u}(e)^m \boldsymbol{x}_i}{\sum_{i=1}^{n} w_{i,u}(e)^m}$$

End loop on centroids, u.
3) Calculation of the total distance d^{total} :

$$d^{total} = \sum_{u=1}^{k} \sum_{i=1}^{n} w_{i,u}(e)^m d(\boldsymbol{x}_i, \boldsymbol{c}_u(e+1)).$$

End loop on epochs e

Fuzzy clustering

Example: motorcycle data set.

- As the dataset is made up categorical variables, the matrix X is the matrix of coordinates of policies, $\boldsymbol{x}_i$, in the weighted Burt space:

$$\boldsymbol{x}_i = D_{i,.} B' / I \quad \text{for } i = 1, ..., n$$

- We run the batch K-means clustering (20 runs). With only 15 clusters, we achieve a Deviance of 5935 (GLM : 5762)

```
import skfuzzy as fuzz #install scikit-fuzzy project
n_clus = 15
cntr, u, u0, d, jm, p, fpc = fuzz.cluster.cmeans(X,n_clus,
m=1.2, error=0.005, maxiter=1000, init=None,seed=40)
```

Fuzzy clustering

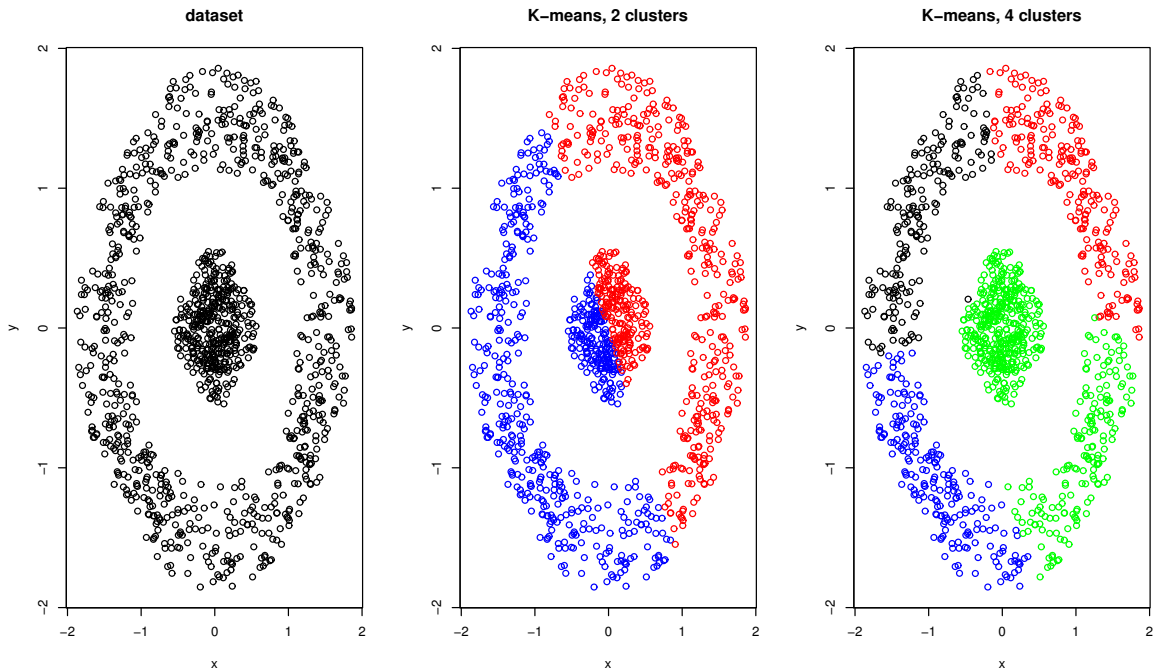
Cluster	Gender	Zone	Class	Owners Age	Vehicule Age	Frequency
0	M	2.0	3.0	40 - 49	5 - 10	0.0133
1	K	4.0	3.0	40 - 49	0 - 4	0.0113
2	M	3.0	4.0	40 - 49	11 - 20	0.0087
3	K	4.0	3.0	40 - 49	11 - 20	0.0064
4	M	4.0	5.0	40 - 49	11 - 20	0.0044
5	M	4.0	3.0	40 - 49	21 -	0.0042
6	M	1.0	4.0	40 - 49	0 - 4	0.0312
7	M	4.0	1.0	40 - 49	21 -	0.0025
8	M	4.0	3.0	40 - 49	5 - 10	0.0078
9	M	3.0	3.0	40 - 49	0 - 4	0.0179
10	M	3.0	6.0	20 - 29	5 - 10	0.0414
11	M	3.0	5.0	40 - 49	11 - 20	0.0098
12	M	2.0	5.0	20 - 29	0 - 4	0.0379
13	M	4.0	3.0	40 - 49	11 - 20	0.0036
14	M	4.0	3.0	40 - 49	0 - 4	0.0093

Roadmap

- Principal Component Analysis
- Factorial component Analysis
- K-means clustering & k-means++
- Clustering with batch k-means*
- Fuzzy clustering
- **Spectral clustering**

Spectral clustering

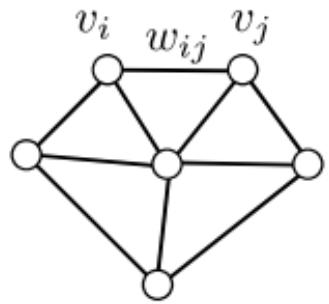
- As the k-means algorithm relies on the Euclidean distance, it fail to identify non convex clusters. For instance:



- The algorithm fails to identify the inner and outer rings.

Spectral clustering

One solution to cluster non-convex shape consists to exploit a deeper data geometry using a graph to represent the dataset.



A graph G is defined by three elements:

- vertices v_i representing data points, $V = \{v_i\}_{i=1}^{n}$.
- edges e_{ij} representing link between vertices, $E = \{e_{i,j} : v_i \longleftrightarrow v_j\}$
- weights w_{ij} (distance between 2 vertices), $W = \{(w_{ij} : w_{ij} \neq 0 \text{ if } v_i \longleftrightarrow v_j\}$.

Spectral clustering

- Elements E and W can be represented as $n \times n$ matrices.
- The elements $e_{i,j}$ of E are equal to 1 if $v_i \longleftrightarrow v_j$ and 0 otherwise.
- The matrix W contains distance w_{ij} if two vertices i and j are linked by an edge.

Laplacian of a graph

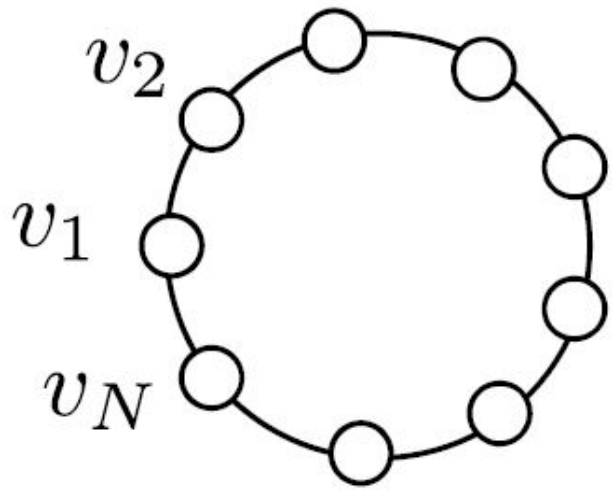
The Laplacian representation of a graph G is defined as:

$$L = D - W$$

where D being a diagonal matrix with diagonal elements that are $D_{ii} = \sum_j w_{i,j}$. D is often referred as the degree matrix.

Spectral clustering*

- Why is matrix L called Laplacian? We can define a function on a graph, $f : V \to \mathbb{R}$ such that $v_i \to f(v_i)$.
- Let us consider a discrete periodic function which takes N values, at times $1, 2, ..., N$. The loop on periods may be represented by a ring graph.



Spectral clustering*

The matrix of edges and weights is in this particular case

$$E = W = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 & 1 \\ 1 & 0 & 1 & 0 & \ddots & 0 \\ 0 & 1 & 0 & 1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 & 0 & 1 \\ 1 & 0 & \dots & 0 & 1 & 0 \end{pmatrix}.$$

If we denote by $\boldsymbol{f} = (f(v_j))_{j=1,\dots,N}$ the vector of values of $f(.)$ at vertices, the product $\boldsymbol{Lf}$ correspond to the second finite difference derivative of the function $f(.)$.

Spectral clustering*

Indeed, a quick calculation reveals that

$$L = \begin{pmatrix} 2 & -1 & 0 & \dots & 0 & -1 \\ -1 & 2 & -1 & 0 & \ddots & 0 \\ 0 & -1 & 2 & -1 & 0 & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & -1 & 2 & -1 \\ -1 & 0 & \dots & 0 & -1 & 2 \end{pmatrix}$$

and

$$(L\boldsymbol{f})_j = -\frac{f(v_{j+1}) - 2f(v_j) + f(v_{j-1})}{1}$$

is the opposite of the discrete second order derivative of f.

Spectral clustering

The spectral analysis of L provides useful information about the structure of the graph.

Spectral analysis

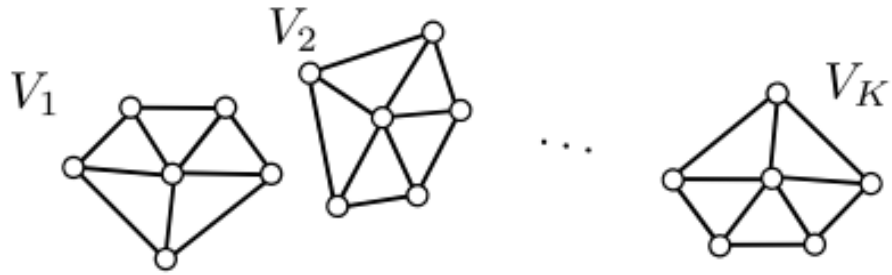
As L is symmetric, it admits the representation

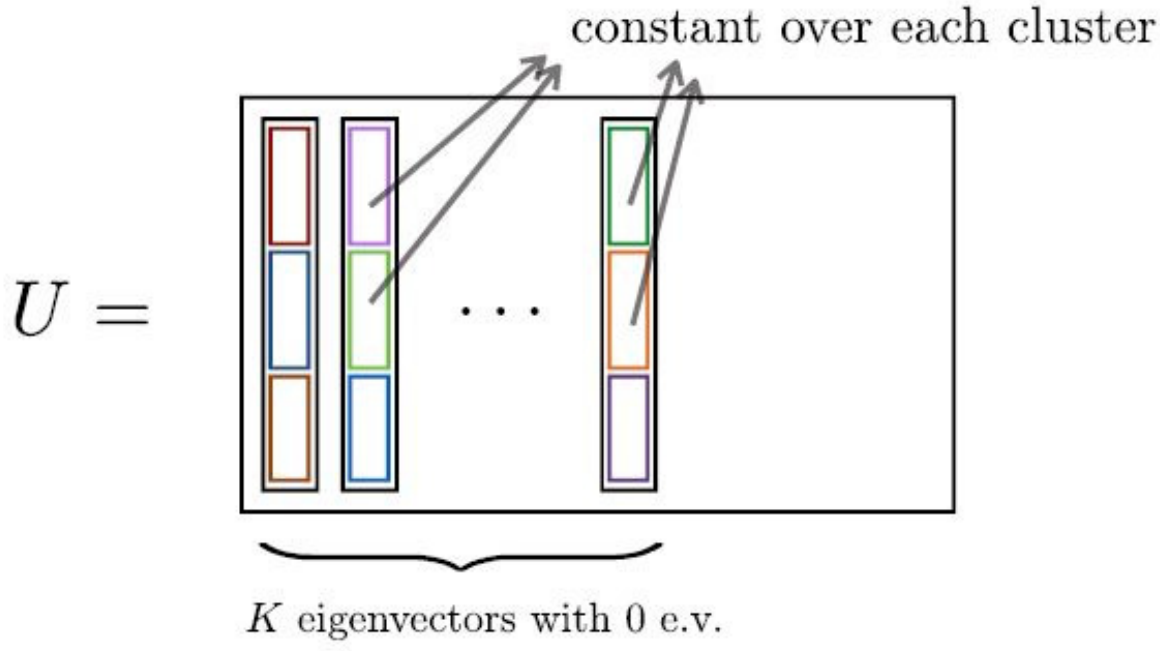
$$L = U \Sigma U^{\top}$$

where U contains eigenvectors and $\Sigma = diag(\lambda_1, ..., \lambda_n)$,diagonal matrix of eigenvalues of L.

- if all vertices of a graph are completely disconnected, all eigenvalues are null.
- if a graph is made of K sub-graphs, we can prove that elements of the K eigenvectors with null eigenvalues are constant over each cluster,

Spectral clustering





We can then run the k-means algorithm with rows of the first K eigenvectors as input representative of vertices.

Spectral clustering

Initialization:

Represent the dataset X as a graph $G = (V, E, W)$

Main procedure:

1) Calculation of the $n \times n$ Laplacian matrix

$$L = D - W$$

2) Compute eigenvectors-values matrix U and Σ of L

$$L = U \Sigma U^{\top}$$

3) Fix k and build the $n \times k$ matrix $U^{(k)}$ of eigenvectors with the k closest eigenvalues to zero

4) Run the k-means algorithmwith the dataset of $U^{(k)}_{i,.}$ for $i = 1, ..., n$.

5) The i^{th} data point is associated to the cluster of $U^{(k)}_{i,.}$

Spectral clustering

- We haven't discussed yet how to represent an initial dataset as a graph. The first step consists to associate vertices $(v_j)_{j=1...n}$ to each data points.
- We next measure the distance between two data points and associate an edge if vertices are "close enough".
- There exist different ways of creating the pairwise similarities graphs representation and this choice generally has an influence on clusters.

Spectral clustering

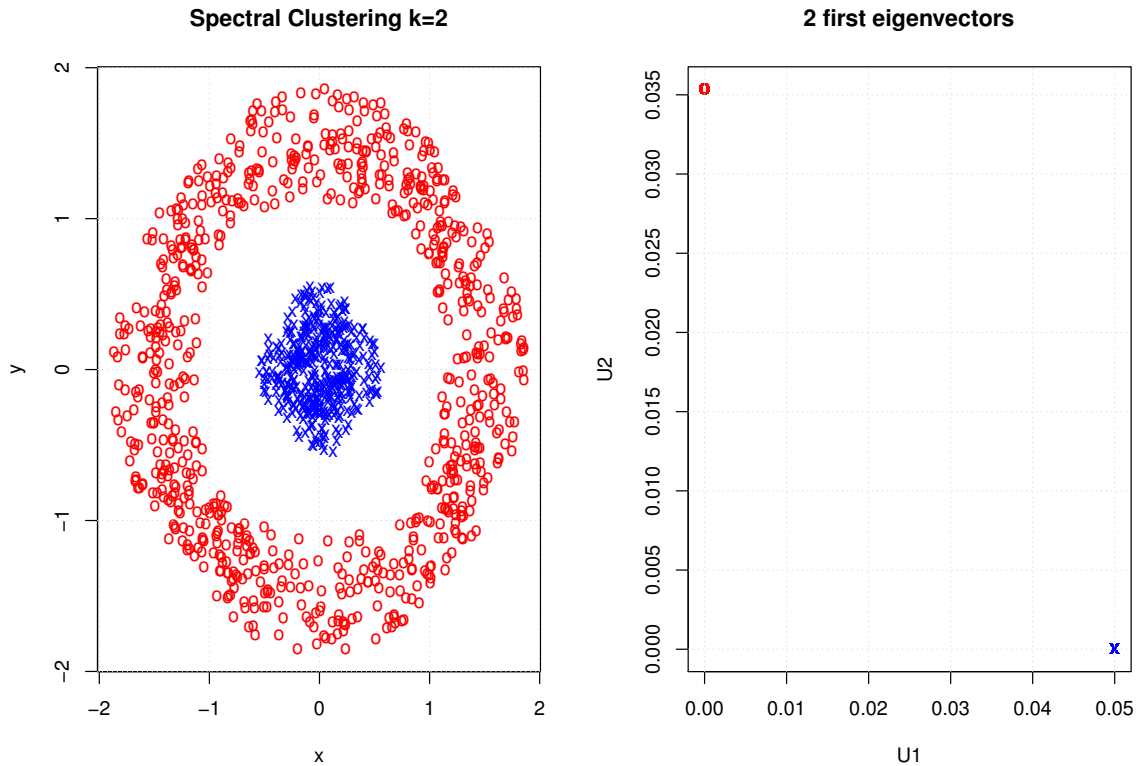
1. **The ϵ-neighborhood graph**: we connect all the points for which the pairwise distances are smaller than ϵ.

2. **The k-nearest neighbor graph**: we connect the vertex v_i with the vertex v_j if v_j is among the k-nearest neighbors of v_i. However, the neighborhood relationship is not symmetric.Two ways to force the symmetry:
   - 2.1 ignore the directions of the edges: we connect v_i and v_j if v_i is among the k-nearest neighbors of v_j **or** if v_j is among the k-nearest neighbors of v_i.
   - 2.2 connect vertices v_i and v_j if both v_i is among the k-nearest neighbors of v_j **and** v_j is among the k-nearest neighbors of v_i. The resulting graph is called the mutual k-nearest neighbors graph.

3. **The fully connected graph**

Spectral clustering

Left plot: partition of a non-convex dataset with spectral clustering. Right plot: pairs of eigenvector coordinates $(U_{i,1}, U_{i,2})$ for $i = 1$ to n.



Spectral clustering

Example: motorcycle data set.

- As the dataset is made up categorical variables, the matrix X is the matrix of coordinates of policies, $\boldsymbol{x}_i = D_{i,.}B' \,/\, I$, in the weighted Burt space.
- Preliminary reduction of the dataset with k-means to 1000 small clusters
- We run the spectral clustering to identify 15 clusters in the reduced dataset
- Graph of 1000 clusters is built with k-nearest neighbor graph (50 neighbors)
- Deviance: 5916 (GLM : 5762)

```
clustering = SpectralClustering(n_clusters=15, n_init = 20,
affinity = 'nearest_neighbors', n_neighbors = 50,
assign_labels = 'discretize', random_state = 42)
```

Spectral clustering

Cluster	Gender	Zone	Class	Owners Age	Vehicule Age	Frequency
0	M	2.0	3.0	50 - 59	5 - 10	0.0093
1	M	1.0	3.0	20 - 29	0 - 4	0.0354
2	K	4.0	6.0	20 - 29	5 - 10	0.0138
3	K	3.0	5.0	30 - 39	11 - 20	0.0068
4	M	4.0	4.0	50 - 59	11 - 20	0.0064
5	M	3.0	6.0	20 - 29	5 - 10	0.041
6	M	4.0	1.0	30 - 39	21 -	0.0021
7	M	4.0	4.0	40 - 49	21 -	0.0061
8	K	1.0	3.0	40 - 49	0 - 4	0.0132
9	M	3.0	5.0	50 - 59	11 - 20	0.0101
10	K	4.0	1.0	30 - 39	21 -	0.0032
11	M	4.0	3.0	40 - 49	11 - 20	0.0048
12	M	1.0	3.0	30 - 39	5 - 10	0.0302
13	K	4.0	3.0	40 - 49	11 - 20	0.0069
14	M	3.0	2.0	10 - 19	5 - 10	0.0163

Conclusions

- Clustering techniques are powerful machine learning technique to uncover hidden patterns in dataset.
- At the era of big data, these methods offer a wide range of tools that are popular amongst the IT community (e.g. in web technologies or signal analysis). For instance we can cite DBSCAN, BIRCH, OPTICS, Gaussian Clustering.
- Nevertheless, clustering techniques are nevertheless under-exploited in actuarial sciences, mainly because they need to be adapted to insurance specific features.