Institut National des Télécommunications et des Technologies de l'Information et de la Communication INT-TIC d'Oran Algérie

Analyse de Données

3ème année ingénieur

Fiche de TP N° 4

Méthodes Descriptives Multi-variées

Analyse Factorielle des Correspondances

Nous avons vu dans le TP 3 que le test d'indépendance de chi 2 permet de vérifier si deux variables qualitatives sont dépendantes ou indépendantes. Ce test se base sur la comparaison du tableau de contingences (tableau qui croise les modalités des deux variables en calculant leurs probabilités jointes) avec un tableau théorique qui aurait été obtenu si les deux variables étaient totalement indépendantes.

Dans un contexte semblable, l'**analyse factorielle des correspondances** (**AFC**) permet d'étudier les liens qui existent entre les modalités des deux variables. Pour cela, elle repose sur une analyse du nuage de points représentant les modalités d'une variable par rapport à l'autre (**profils lignes** ou **profils colonnes**).

Le tableau des profils lignes (resp. des profils colonnes) est celui des probabilités conditionnelles d'avoir une modalité de la 2ème variable (resp. de la 1ère variable) en fixant une modalité de la 1ère variable (resp. de la 2ème variable). Il s'obtient facilement en divisant le tableau de contingences (les probabilités conjointes) par les probabilités marginales des modalités de la 1ère variable (resp. de la 2ème variable). Dans le cas d'un tableau de profils lignes, chaque ligne est considérée comme un point (un individu) et chaque colonne est considérée comme un axe (une variable).

A partir de ce nuage de points, des **composantes principales** (les **facteurs,** d'où l'appellation **analyse factorielle**) sont calculées selon le même principe de l'ACP, mais en tenant compte d'une distance particulière entre les points du nuage : distance de chi2.

Le nuage de points initial ainsi que les axes initiaux (donc toutes les modalités des deux variables) sont ensuite projetés sur les composantes principales (généralement le 1er plan principal). La proximité entre deux modalités sur le plan principal est traduite par un lien entre les deux modalités.

Ces aspects seront illustrés dans ce TP à travers un exemple.

Le fichier de données utilisé dans ce TP est `students-parents.csv`, qui résume la répartition des spécialités d'études de 10000 étudiants et de la catégorie socio-professionnelles de leurs parents.

Les catégories socio-professionnelles des parents (modalités de la 1ère variable) sont au nombre de 9 : Expert Agricole, Salarié, Patron, Profession Libérale, Cadre Moyen, Employé, Ouvrier, Service et Autre.

Les spécialités d'études des étudiants (modalités de la 2ère variable) sont au nombre de 8 : Droit, Sciences Economiques, Lettres, Sciences, Médecine, Pharmacie, Pluridisciplinaire et IUT.

Les modules utilisés sont les mêmes utilisés dans les TPs précédents : `numpy`, `pandas` et `matplotlib`.

```
>>> import numpy as np
>>> import pandas as pd
>>> import matplotlib.pyplot as plt
>>> import os
```

Importation et préparation des données

```
>>> os.chdir('C:\\...\\...\\DataExamples')
>>> data = pd.read_csv('students-parents.csv', sep=',')

>>> data
               Droit  Sc Eco  Lettres  ...  Pharmacie  Pluridisc  IUT
Exp agricole      80      36      134  ...         28         11   58
Salarié            6       2       15  ...          1          1    4
Patron           168      74      312  ...         53         21   62
Prof liberale    470     191      806  ...        164         45   79
Cadre moyen      236      99      493  ...         56         36   87
Employé          145      52      281  ...         30         20   54
Ouvrier          166      64      401  ...         23         28  129
Service           16       6       27  ...          2          2    8
Autre            305     115      624  ...         47         42   90

>>> n, m = data.shape                          # Nombre de modalités des 2 variables
>>> n
9
>>> m
8
```

Nous calculons ensuite la matrice `P` des probabilités conjointes, et les probabilités marginales `P_i` et `P_j` :

```
>>> P = data.apply(lambda x:x/data.sum().sum())
>>> P
                Droit  Sc Eco  Lettres  ...  Pharmacie  Pluridisc     IUT
Exp agricole   0.0080  0.0036   0.0134  ...     0.0028     0.0011  0.0058
Salarié        0.0006  0.0002   0.0015  ...     0.0001     0.0001  0.0004
Patron         0.0168  0.0074   0.0312  ...     0.0053     0.0021  0.0062
Prof liberale  0.0470  0.0191   0.0806  ...     0.0164     0.0045  0.0079
Cadre moyen    0.0236  0.0099   0.0493  ...     0.0056     0.0036  0.0087
Employé        0.0145  0.0052   0.0281  ...     0.0030     0.0020  0.0054
Ouvrier        0.0166  0.0064   0.0401  ...     0.0023     0.0028  0.0129
Service        0.0016  0.0006   0.0027  ...     0.0002     0.0002  0.0008
Autre          0.0305  0.0115   0.0624  ...     0.0047     0.0042  0.0090
```

```
>>> P_j = P.sum()                          # Somme des colonnes
>>> P_j
Droit         0.1592
Sc Eco        0.0639
Lettres       0.3093
Sciences      0.1490
Medicine      0.2005
Pharmacie     0.0404
Pluridisc     0.0206
IUT           0.0571
dtype: float64


>>> P_i = P.T.sum()                        # Somme des lignes
>>> P_i
Exp agricole     0.0511                   # P.T est la transposée de P
Salarié          0.0039
Patron           0.1035
Prof liberale    0.3031
Cadre moyen      0.1552
Employé          0.0850
Ouvrier          0.1131
Service          0.0080
Autre            0.1771
dtype: float64


>>> P_i.sum()                              # Vérifier que la somme est égale à 1
1.0
>>> P_j.sum()                              # Vérifier que la somme est égale à 1
1.0
```

Nous pouvons ensuite calculer le tableau des profils lignes (probabilités conditionnelles, spécialité d'études sachant la catégorie socio-professionnelles du père) :

```
>>> line_profiles = (P.T / P_i).T
>>> line_profiles
                  Droit    Sc Eco   Lettres  ...  Pharmacie  Pluridisc       IUT
Exp agricole   0.156556  0.070450  0.262231  ...   0.054795   0.021526  0.113503
Salarié        0.153846  0.051282  0.384615  ...   0.025641   0.025641  0.102564
Patron         0.162319  0.071498  0.301449  ...   0.051208   0.020290  0.059903
Prof liberale  0.155064  0.063016  0.265919  ...   0.054108   0.014847  0.026064
Cadre moyen    0.152062  0.063789  0.317655  ...   0.036082   0.023196  0.056057
Employé        0.170588  0.061176  0.330588  ...   0.035294   0.023529  0.063529
Ouvrier        0.146773  0.056587  0.354553  ...   0.020336   0.024757  0.114058
Service        0.200000  0.075000  0.337500  ...   0.025000   0.025000  0.100000
Autre          0.172219  0.064935  0.352343  ...   0.026539   0.023715  0.050819
```

Nous devons avoir une somme des lignes égale à 1 :

```
>>> line_profiles.T.sum()
Exp agricole     1.0
Salarié          1.0
Patron           1.0
Prof liberale    1.0
Cadre moyen      1.0
Employé          1.0
Ouvrier          1.0
Service          1.0
Autre            1.0
dtype: float64
```

Calcul de la matrice d'inertie

Dans le tableau `line_profiles`, les modalités en lignes forment un nuage de points et les modalités en colonnes forment les axes de l'espace (les variables). La covariance $V_{jj'}$ entre deux variables j et j' (deux colonnes) est donnée par la formule :

$$V_{jj'} = \sum_i \frac{p_{ij}p_{ij'}}{\sqrt{p_{\cdot j}}\sqrt{p_{\cdot j'}}p_{i\cdot}} - \sqrt{p_{\cdot j}}\sqrt{p_{\cdot j'}}$$

où p_{ij} sont les probabilités conjointes et où $p_{\cdot j}$, $p_{\cdot j'}$ et $p_{i\cdot}$ sont les probabilités marginales des modalités colonnes j et j' et de la modalité ligne i.

Les covariances $V_{jj'}$ forment la matrice de variance-covariance V, dont les vecteurs propres sont les composantes principales recherchées. Il est démontré que les vecteurs propres de cette matrice équivalent aux vecteurs propres de la matrice d'inertie S, dont les termes sont calculés de la façon suivante :

$$s_{jj'} = \sum_i \frac{p_{ij}p_{ij'}}{\sqrt{p_{\cdot j}}\sqrt{p_{\cdot j'}}p_{i\cdot}}$$

C'est la matrice S que nous allons utiliser pour le calcul des composantes principales :

- Les termes p_{ij} sont les éléments de notre matrice `P`.
- Les termes $p_{\cdot j}$ et $p_{\cdot j'}$ sont les éléments de notre vecteur `P_j`.
- Les termes $p_{i\cdot}$ sont les éléments de notre vecteur `P_i`.

Pour cela, nous allons calculer une matrice intermédiaire `tmp` (de calcul) dont les éléments représentent les termes $\frac{p_{ij}}{\sqrt{p_{\cdot j}}}$:

```
>>> tmp = P / np.sqrt(P_j)
```

La matrice S s'obtient de la façon suivante :

```
>>> S = np.dot( (tmp.T / P_i)  ,  tmp)
```

Calcul des vecteurs propres et des valeurs propres

```
>>> eigen_vals, eigen_vecs = np.linalg.eig(S)
>>> eigen_vals
array([1.00000000e+00, 3.97371637e-02, 5.45168965e-03, 1.12968930e-03,
       9.60397847e-04, 1.33235306e-04, 5.35610772e-05, 1.05719860e-06])
```

Nous remarquons que la première valeur propre est égale à 1 (ce qui est toujours le cas). Elle correspond à une valeur propre nulle dans le cas d'une matrice `V` de variance-covariance. Cette valeur propre, ainsi que le vecteur propre correspondant, seront ignorés dans les étapes d'analyse suivantes, puisque les projections sur cette composante sont inutiles (même point) :

```
>>> eigen_vals = eigen_vals[1:]
>>> eigen_vecs = eigen_vecs[:,1:]
```

Comme dans une ACP, on procède au tri décroissant des valeurs propres :

```
>>> index = np.argsort(eigen_vals)[::-1]
>>> index
array([0, 1, 2, 3, 4, 5, 6], dtype=int64)
>>> eigen_vals = eigen_vals[index]
>>> eigen_vecs = eigen_vecs[:,index]
```

Nous calculons ensuite le pourcentage de chaque valeur propre, qui correspond au pourcentage d'inertie de chaque composante :

```
>>> pourcentage = eigen_vals/eigen_vals.sum()*100
>>> pourcentage
array([8.37157101e+01, 1.14852704e+01, 2.37995702e+00, 2.02330464e+00,
       2.80691605e-01, 1.12839045e-01, 2.22723827e-03])
```

On remarque que les deux premières composantes résument respectivement 83.71 et 11.48 de l'inertie totale. Les deux premières composantes seront donc retenues pour la projection des modalités.

Calcul de la projection des modalités lignes et colonnes

Nous allons dans ce qui suit calculer les coordonnés des modalités lignes et colonnes dans le premier plan factoriel. Les formules de projection sont les suivantes :

- Pour un point i du nuage (modalité ligne), la projection F_{ci} sur la composante principale p est donnée par : $$F_{ci} = \sum_j U_{cj} \frac{p_{ij}}{p_{i.}\sqrt{p_{.j}}}$$
- Pour une variable j (modalité colonne), la projection F_{cj} sur la composante principale c est donnée par : $$F_{cj} = \sqrt{\lambda_c} \frac{U_{cj}}{\sqrt{p_{.j}}}$$

Où U_{cj} représentent l'élément j du vecteur propre c (composante principale c) et où λ_c représente la valeur propre associée au vecteur propre c.

Nous allons nous servir des matrices et vecteurs calculés précédemment :

```
>>> coordinates_i = np.dot(tmp ,eigen_vecs[:,:2])
>>> coordinates_i = (coordinates_i.T / np.asarray(P_i)).T
>>> coordinates_i
array([[ 0.23278822, -0.2263911 ],            # Coordonnés des modalités lignes
       [ 0.32277807,  0.03299944],
       [-0.01971079, -0.0298329 ],
       [-0.2635195 , -0.02385993],
       [ 0.04955325,  0.01144584],
       [ 0.10347731,  0.03316551],
       [ 0.33484769, -0.02992433],
       [ 0.29118275,  0.02117102],
       [ 0.06816229,  0.11507143]])
```

```
>>> coordinates_j = np.sqrt(eigen_vals[:2]) * eigen_vecs[:,:2]
>>> coordinates_j = (coordinates_j.T / np.sqrt(np.asarray(P_j))).T
>>> coordinates_j
array([[ 0.00354832,  0.03265875],
       [-0.01147977, -0.01157534],            # Coordonnés des modalités colonnes
       [ 0.09028226,  0.07425756],
       [ 0.09556202, -0.0502928 ],
       [-0.3133872 , -0.02020306],
       [-0.24901539, -0.1810872 ],
       [ 0.17741699,  0.06280885],
       [ 0.47714859, -0.17269829]])
```

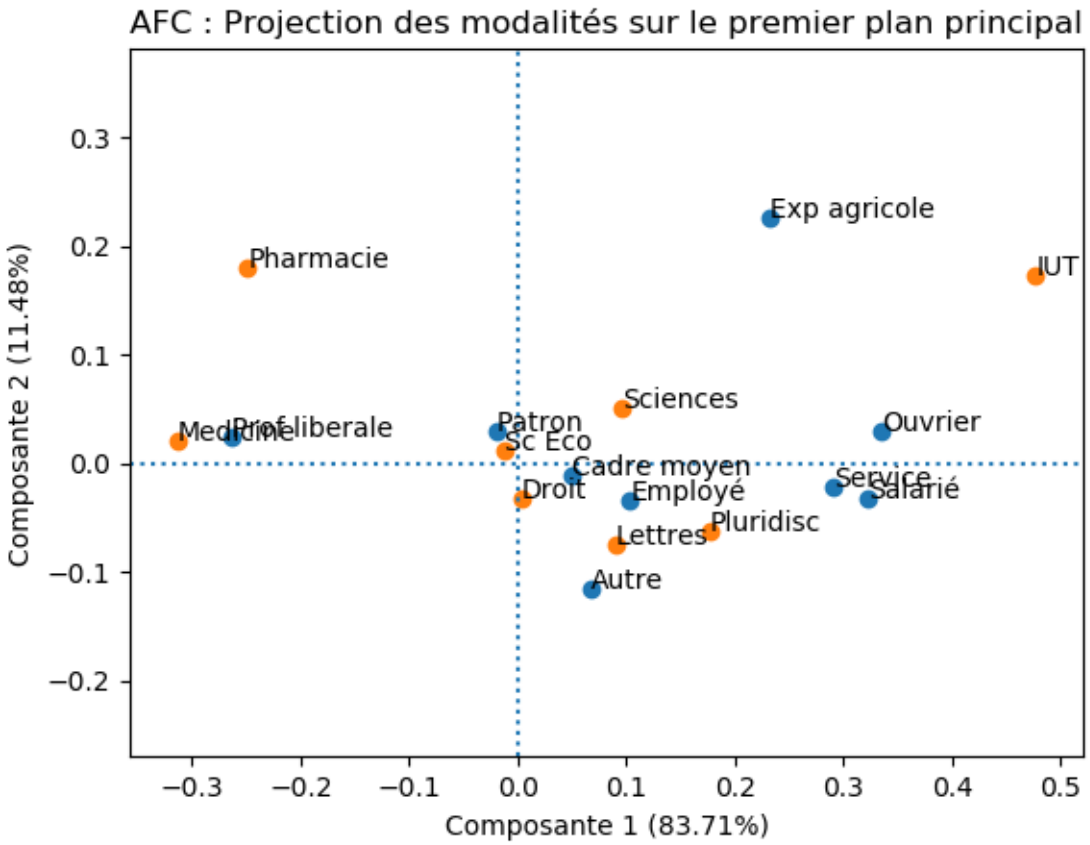
Visualisation

Dans le graphique de visualisation, nous avons besoin des noms de toutes les modalités. Le code suivant permet de récupérer ces noms :

```
>>> modalities_i = np.asarray(data.index)
>>> modalities_i
array(['Exp agricole', 'Salarié', 'Patron', 'Prof liberale',
       'Cadre moyen', 'Employé', 'Ouvrier', 'Service', 'Autre'],
      dtype=object)

>>> modalities_j = np.asarray(data.columns)
>>> modalities_j
array(['Droit', 'Sc Eco', 'Lettres', 'Sciences', 'Medicine', 'Pharmacie',
       'Pluridisc', 'IUT'], dtype=object)
```

Le code suivant permet de visualiser la projection de ces modalités sur le 1er plan factoriel :

```
>>> plt.axis('equal')

>>> plt.axvline(0, ls=':'), plt.axhline(0, ls=':')

>>> plt.scatter(coordinates_i[:,0] , coordinates_i[:,1])

>>> plt.scatter(coordinates_j[:,0] , coordinates_j[:,1])

>>> for i in range(modalities_i.size):
...  plt.annotate(modalities_i[i] , (coordinates_i[i,0] , coordinates_i[i,1]))
...

>>> for j in range(modalities_j.size):
...  plt.annotate(modalities_j[j] , (coordinates_j[j,0] , coordinates_j[j,1]))
...

>>> plt.title('AFC : Projection des modalités sur le premier plan principal')

>>> plt.xlabel('Composante 1 (83.7%)')

>>> plt.ylabel('Composante 2 (11.48%)')

>>> plt.show()
```



Selon ce graphique, on peut s'apercevoir que les étudiants de médecine et de pharmacie présentent des profils proches, et que les étudiants dont le père exerce une profession libérale optent généralement pour ces spécialités. Ces étudiants présentent des profils plutôt opposés (selon le 1er facteur) par rapport aux profils des étudiants des IUT. De même, les étudiants dont les parents sont des ouvriers, des salariés, des experts agricoles et des travailleurs de métiers de services, ont des profils opposés (en terme de choix de spécialité) par rapport aux profils des étudiants dont les parents exercent des professions libérales. On peut également remarquer la proximité entre « Patron » et « Sc Eco », qui s'interprète par une préférence des étudiants fils de patrons pour les sciences économiques.

Remarque

Nous avons vu que l'**analyse factorielle des correspondances** (**AFC**) permet de décrire les relations entre deux variables qualitatives.

Pour le cas de plusieurs variables qualitatives, une généralisation de l'AFC existe : c'est la méthode d'**analyse factorielle des correspondances multiples** (**AFCM**).

R. HACHEMI