Organisation des projets

- Vous pouvez travailler par groupes de 1 ou 2 étudiants
- Chaque groupe choisit un sujet différent (les projets libres peuvent être choisis par plusieurs groupes).
- Les sujets comportent une liste de questions, mais si vous avez **vos propres idées** pour analyser les données **suivez les en priorité**.
- Le niveau de difficulté de chaque sujet est indiqué en en-tête. D'un point de vue analyse des données, aucune connaissance autre que celles vues en cours n'est nécessaire. Ne choisissez les sujets faciles (dont moins riches) que si vous avez des difficultés, par exemple en programmation.
- Les données nécessaires aux projets se trouvent dans le folder `students_etudiants`.
- Chaque groupe devra rendre un rapport sur son projet. Le rapport devra être accompagné du code Matlab développé (ne pas rendre de folder compressé, .zip ou .tar par exemple, car ils seraient rejetés par notre mailer). Si le groupe comprend 2 étudiants, un paragraphe expliquera comment le travail a été organisé et les tâches réparties. Le rapport comporte une analyse aussi complète que possible des résultats.

Big data

1. **ACP : bilan des entreprises françaises en 2013** difficulté : assez élevée

   Objectifs Le but de ce projet est d'étudier le bilan des entreprises françaises en 2013 en fonction de leur secteur d'activité. Les données officielles viennent d'être rendues disponibles par l'Etat, sous forme d'un fichier .xls "bilan au niveau sous-classe", sur le site de l'INSEE :

   http://www.insee.fr/fr/themes/detail.asp?reg_id=0&ref_id=esane-2013

   Les valeurs du fichier .xls ont été recopiées dans le fichier ascii `bilan_X.txt` pour simplifier leur lecture. Les noms des secteurs se trouvent dans le fichier `bilan_secteurs.txt`. Les noms des caractéristiques comptables(*) sont dans le fichier `bilan_caracteristiques.txt`. L'objectif est de tirer le maximum d'informations pertinentes de ces données.

   Préparation des données Ecrire un programme effectuant les tâches suivantes :

   - Charger (fonction `load`) dans Matlab, les données du fichier `bilan_X.txt`.
   - Comme souvent, de nombreuses données sont manquantes dans le fichier (marquées par des -1).
     - Dans un premier temps, supprimez toutes les lignes de `X` dans lesquelles il manque des données et passez au • suivant.

  - Quand vous aurez complètement terminé l'étude de la matrice dans laquelle toutes les lignes incomplètes ont été supprimées reprenez l'étude en ne supprimant que les secteurs pour lesquels il y a plus qu'une donnée manquante. Remplacer les valeurs manquantes par une valeur raisonnable, par exemple une valeur moyenne. Ecrire un code qui remplace les -1 restants par des valeurs qui vous semblent raisonnables.

- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X` (vous n'êtes pas obligés de les utiliser). La transposée de la matrice `M` est donnée par `M'`.

ACP

- Pour calculer les valeurs propres et les vecteurs propres de `M`, vous pouvez utiliser la fonction Matlab `[E,D]=eig(M)`. La matrice `E` contient les vecteurs propres (en colonne) et la matrice `D` est diagonale avec pour composantes les valeurs propres de `M`. Si les valeurs propres ne sont pas classées par ordre décroissant, vous pouvez utiliser `fliplr` et `flipud`, pour réordonner les matrices.
- A l'aide de ces outils, effectuer l'ACP.

Analyses

- Certaines variables sont très corrélées pour des raisons triviales (comme `Tab` et `Tan`). Cherchez si des transformations de ces variables ne pourraient pas apporter des informations nouvelles permettant de caractériser les secteurs les uns par rapport aux autres.
- Exemples d'analyses possibles :
  - A l'aide du nom des secteurs, vous pouvez tracer d'une couleur différentes les points relatifs à un groupe de secteurs donné (par exemple, les secteurs relatifs à la 'Reparation').
  - Vous pouvez chercher la nature des secteurs ayant un comportement extrême.
  - etc. (essayez vos idées en priorité).
- Analysez les résultats des ACP que vous aurez obtenues avec différentes versions de la matrice `X` et tirez-en des conclusions à la fois méthodologique et sur le bilan des secteurs économiques en 2013.

(*) Les caractéristiques comptables sont les suivantes:

(a) `Nul` Nombre d'unites légales
(b) `Cna` Capital souscrit non appelé
(c) `Imi` Immobilisations incorporelles
(d) `Imc` Immobilisations corporelles
(e) `Ter` Terrains
(f) `Con` Constructions
(g) `Itm` Installations techniques, materiel et outillage industriels
(h) `Aic` Autres immobilisations corporelles
(i) `Mdt` dont materiel de transport :
(j) `Ime` Immobilisations en cours
(k) `AeA` Avances et acomptes
(l) `Imf` Immobilisations financières
(m) `Tai` Total de l'actif immobilise
(n) `Smp` Stocks - Matières premieres approvisionnement et en cours
(o) `Sdm` Stocks de marchandises
(p) `Aav` Avances et acomptes verses sur commandes
(q) `Ccr` Clients et comptes rattaches
(r) `Acr` Autres créances
(s) `Vmp` Valeurs mobilières de placement
(t) `Dis` Disponibilité
(u) `Cdr` Comptes de régularisation - Charges constatées d'avances
(v) `Tac` Total de l'actif circulant
(w) `Acr` Autres comptes de régularisation
(x) `Tab` Total actif brut
(y) `Tan` Total de l'actif net des amortissements et provisions inscrits a l'actif

2. **ACP : fluctuations boursières** difficulté : moyenne

Objectifs Le but de ce projet est de construire un codeur ACP et son décodeur pour compresser des courbes indiquant les fluctuations de la valeur de différentes actions. Nous étudierons dans quelles conditions l'information importante est conservée en fonction du taux de compression de ces courbes. On étudie 70 événements boursiers pendant lesquels la valeur d'une action grimpe brusquement, comme une marche d'escalier. Ces brusques valorisations entraînent souvent des augmentations d'autres actions du même secteur. Ces augmentations, dites induites, sont suivies de rapides re-descentes sur le niveau précédant l'évènement. Pour chacun de ces évènements on donne deux courbes et un indice:

- `Escalier` : variations temporelles de la valeur de l'action d'une entreprise subissant une "marche d'escalier". L'amplitude de la marche a été normalisée.
- `Induit` : variations temporelles de l'action d'une autre entreprise du même secteur subissant une augmentation induite.
- `Ip` : un indice qui mesure la proximité sectorielle entre les deux entreprises.

L'expérience passée a montré que l'amplitude de la courbe induite est très corrélée à l'indice de proximité. Une bonne approximation de Ip est donnée par la formule suivante:

$$Ip_{\text{app}} = \sum_{j=1}^{25} \text{Induit}_j^2$$

(Ip est la valeur vraie de l'indice de proximité, Ip_{app} est sa valeur approchée obtenue grace à la variation induite). Les courbes `Escalier` et `Induit` vont être compressées grace à l'ACP (pour faciliter leur transmission et leur mémorisation). Elles seront ensuite décompressées. On vérifiera de deux manières que l'on n'a pas perdu trop d'information dans le processus de compression/décompression :

- en comparant la forme des courbes après décompression à celle avant compression
- en comparant la valeur de l'indice de proximité après décompression à celui avant compression

Préparation Ecrire un programme Matlab effectuant les taches suivantes:

- Lire les données dans les fichiers ascii `bourse_Escalier.txt`, `bourse_Induit.txt` et `bourse_Ip.txt`.
- Calculer l'indice de proximité Ip_{app}, pour tous les évènements. Visualiser la correlation entre Ip_{app} et Ip.
- Construire la matrice de données `X` à partir des matrices `Escalier` et `induit`. Chaque ligne i de `X` correspond à un évènement, c'est-à-dire aux 25 canaux de la ième fluctuation boursière en escalier suivis par les 25 canaux de la ième fluctuation induite. La taille de la matrice `X` est donc 70×50.

Codage

- Pour calculer les valeurs propres et les vecteurs propres de `M`, vous pouvez utiliser la fonction Matlab `[E,D]=eig(M)`. La matrice `E` contient les vecteurs propres (en colonne) et la matrice `D` est diagonale avec pour composantes les valeurs propres de `M`. Si les valeurs propres ne sont pas classées par ordre croissant, vous pouvez utiliser `fliplr` et `flipud` pour réordonner les matrices.
- A l'aide de ces outils, construire le codeur (compression) et le décodeur (décompression) ACP.

Analyses

- Comparer les courbes initiales avec les courbes après compression/décompression.
- Définir une variable raisonnable pour mesurer le *taux de compression* des données quand toutes les variables principales ne sont pas transmises du codeur au décodeur.
- Etudier l'évolution de ρ_p, le coefficient de corrélation entre Ip et Ip_{app}, en fonction de l'information expliquée par les axes principaux transmis et en fonction du taux de compression des courbes.
- Conclusions.

3. ACP : Chiffre d'Affaire des branches d'activité en 2012 difficulté moyenne

Objectifs Le but de ce projet est d'étudier le chiffre d'affaire des différentes branches de l'industrie françaises en 2012 en fonction de leurs effectifs. Les données officielles viennent d'être rendues disponibles par l'Etat, sous forme d'un fichier .xls "Chiffre d'affaire par branche (niveau groupe) et tranche d'effectifs, en 2012", sur le site de l'INSEE :

`http://www.insee.fr/fr/themes/detail.asp?reg_id=0&ref_id=esane-branche-2012`

Les valeurs du fichier .xls ont été recopiées dans le fichier ascii `CA_2012_X.txt` pour simplifier leur lecture. Les noms des branches se trouvent dans le fichier `CA_2012_branches.txt`. Les tranches d'effectifs sont dans le fichier `CA_2012_effectifs.txt`. L'objectif est de tirer le maximum d'informations pertinentes de ces données.

ACP Ecrire un programme effectuant les taches suivantes:

- Lire les données dans le fichier ascii `CA_2012_X.txt`.
- Les valeurs -1 dans la matrice indiquent des données manquantes. Proposez une ou plusieurs solutions pour continuer l'analyse malgré ces valeurs manquantes.
- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X` (vous n'êtes pas obligés de les utiliser). La transposée de la matrice `M` est donnée par `M'`.
- Pour calculer les valeurs propres et les vecteurs propres de `M`, vous pouvez utiliser la fonction Matlab `[E,D]=eig(M)`. La matrice `E` contient les vecteurs propres (en colonne) et la matrice `D` est diagonale avec pour composantes les valeurs propres de `M`. Si les valeurs propres ne sont pas classées par ordre décroissant, vous pouvez utiliser `fliplr` et `flipud` pour réordonner les matrices.
- A l'aide de ces outils, effectuer l'ACP.

Analyses

- Certaines variables sont très corrélées pour des raisons triviales. Cherchez si des transformations de ces variables ne pourraient pas apporter des informations nouvelles permettant de caractériser les branches les unes par rapport aux autres. Observez les modifications induites sur les informations portées par les axes principaux.
- Analysez les résultats des ACP que vous aurez obtenus avec différentes versions de la matrice `X` et tirez-en des conclusions méthodologiques et sur le bilan des branches économiques en 2012.
- Exemples d'analyses possibles :
  - La plus petite des valeurs propres est nulle. Pouvez vous expliquer pourquoi?
  - A l'aide du nom des branches, vous pouvez tracer d'une couleur différentes les points relatifs à un groupe de branches donné.

– Vous pouvez cherche la nature des branches ayant un comportement extrême.
– etc. (essayez vos idées en priorité).

4. ACP : analyse d'échantillons d'eau industrielle difficulté : simple

Objectifs Un grand groupe de métallurgie souhaite comparer la qualité des eaux utilisées dans ses différentes usines. Des prélèvements sont donc effectués pendant une année. Le but de ce travail est d'analyser, à l'aide de l'ACP, cet ensemble d'échantillons d'eau. Les valeurs présentées dans le fichier ascii `eau_industrielle.txt` sont des moyennes pour chaque usine sur l'année. Les dosages effectués sont : le calcaire (Ca), les sulfates/sulfites (Sul), les polychlorobiphényles (PCB), le mercure (Hg) et une note tenant compte de la qualité globale de l'eau (Glo). L'ordre des villes dans lesquelles se trouvent les usines est le suivant : Rouen, Le Havres, Paris, Troyes, Orléans, Nantes, Angers, Saint-Nazaire, Agen, Bordeaux, Castelsarrasin, Saint-Gaudens, Toulouse, Nîmes, Valence, Lyon, Orange, Marseille, Avignon.

Analyse

- La fonction `load` permet de lire le fichier de données. Codez de deux manières différentes l'ACP du tableau: en utilisant la fonction Matlab `pca` (si la toolbox statistique existe sur votre ordinateur), puis en programmant vous-même les étapes du calcul de l'ACP (vous pouvez utiliser la fonction Matlab `[E,D]=eig(V)` pour calculer les valeurs/vecteurs-propres).
- Visualiser les résultats (fonction Matlab `plot`).
- En tirer le maximum de conclusions.
- Un échantillon unique à été prélevé dans l'usine de Tain l'Hermitage, on n'a donc pas souhaité l'inclure dans le calcul des axes principaux. Par contre, on le traite comme un échantillon passif. Ses valeurs sont : Ca = 0.8422, Sul = 1.7390, PCB = 0.8712, Hg = 0.1800, Glo = 0.6903. Calculer sa position dans le plan principal. Conclusions ?

5. ACP : evolutions du bilan des unités légales entre 2010 et 2013 difficulté : moyenne

Objectifs Le but de ce projet est d'étudier l'évolution du bilan des unités légales françaises entre 2010 et 2013 en fonction de leur secteur d'activité. La comparaison s'effectuera en utilisant les résultats 2013 comme données actives et les résultats de 2010 à 2012, comme des données passives. Les données officielles viennent d'être rendues disponibles par l'Etat, sous forme de fichiers .xls, sur le site de l'INSEE :

http://www.insee.fr/fr/bases-de-donnees/default.asp?page=presentation-stat-annuelle-entreprise.htm

(voir le paragraphe **2-Données détaillées sur les unités légales**)

Les valeurs des fichiers .xls ont été recopiées dans les fichiers ascii `entreprises_X_année.txt` pour simplifier leur lecture. La colonne "Effectifs occupés" a été supprimée car elle n'était pas présente toutes les années. Les noms des secteurs se trouvent dans le fichier `entreprises_secteurs.txt`. Les noms des caractéristiques comptables sont dans le fichier `entreprises_caracteristiques.txt`. L'objectif est de tirer le maximum d'informations pertinentes de ces données et d'observer leurs évolutions sur quatre ans.

Traitement des données Ecrire un programme effectuant les taches suivantes:

- Lire les données dans le fichier ascii `entreprises_X_année.txt`.
- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X` (vous n'êtes pas obligés de les utiliser). La transposée de la matrice `M` est donnée par `M'`.
- Pour calculer les valeurs propres et les vecteurs propres de `M`, vous pouvez utiliser la fonction Matlab `[E,D]=eig(M)`. La matrice `E` contient les vecteurs propres (en colonne) et la matrice `D` est diagonale avec pour composantes les valeurs propres de `M`. Si les valeurs propres ne sont pas classées par ordre décroissant, vous pouvez utiliser `fliplr` et `flipud`, pour réordonner vos matrices.
- A l'aide de ces outils, effectuer l'ACP des données pour l'année 2013.

Analyses des résultats

- Les variables sont très corrélées pour des raisons triviales. Cherchez si une transformation simple de ces variables ne permettrait pas de mieux comparer les secteurs entre eux en supprimant une certaine très forte corrélation triviale. Observez l'effet de cette modification sur les informations portées par les axes.
- Analysez les résultats des ACP que vous aurez obtenus avec différentes versions de la matrice `X` et tirez-en des conclusions méthodologiques et sur le bilan des secteurs économiques en 2013.
- traitez les tableaux relatifs aux années 2010 à 2012 comme des données passives. Quelles conclusions sur l'évolution des bilans des unités légales ?

6. ACP : œil électronique difficulté : assez élevée

Objectifs Les cadres de Disney World Paris veulent connaître la proportion d'enfants et d'adultes accompagnants qui montent dans une attraction donnée. Ils ont placé un œil électronique à l'entrée. Le signal électrique délivré par l'œil (Fig. 1) est sensible à la taille et à la vitesse de déplacement de la personne. Le but de ce projet est de voir dans quelle mesure ces signaux peuvent être compactés, pour accélérer leur traitement et minimiser l'espace mémoire nécessaire pour les conserver, tout en continuant de permettre d'en déduite l'âge (la taille) des visiteurs. l'ACP sera utilisée pour conserver le maximum d'information à l'aide d'un nombre réduit de variables principales. Des signaux typiques sont présentés Figure 1.

On définit x comme la somme des intégrales dans les portes Prompt et Delayed. Elle est proportionnelle à la vitesse de déplacement de la personne. La variable y est l'intégrale dans la porte Delayed. La variable $f = y/x$, est utilisée pour discriminer enfants et parents puisque, à vitesse de déplacement donnée, elle prend des valeurs plus grandes pour les parents que pour les enfants.

Discrimination Ecrire un programme qui réalise les taches suivantes :

- Lire les signaux dans le fichier ascii `signaux.txt` (vous pouvez utiliser la fonction Matlab `load`)
- A l'aide de la fonction Matlab `plot`, visualiser le premier signal puis la superposition de tous les signaux.
- Tracer y en fonction de x .
- Calculer les variables x, y et f pour tous les signaux en plaçant les portes d'intégration de manière similaire à celle de la figure.
- Tracer la distribution de la variable f (vous pouvez utiliser la fonction Matlab `hist`). Commenter les résultats. Optimisez la séparation entre parents et enfants en modifiant la position des portes.
- Le seuil de discrimination entre enfants et parents est placé au minimum entouré par les deux maxima de la distribution. Evaluer, aussi bien que possible, la proportion de visiteurs mal identifiés p_{mi}.

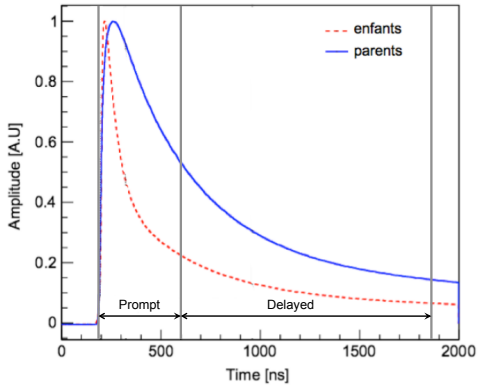

Figure 1: "Prompt" et "Delayed" sont deux portes d'intégration du signal. Pour des enfants (taille petite) et des parents (taille élevée) se déplaçant à la même vitesse, les sommes des intégrales dans les portes Prompt et Delayed sont similaires. Par contre, l'intégrale dans la seule porte Delayed et plus importante pour les adultes. Sur cette figure, les différences de forme entre les signaux enfants et parents ont été accentuées.

Codage

- La transposée de la matrice `M` est donnée par `M'`.
- La fonction Matlab `[E,D]=eig(M)` calcule les vecteurs/valeurs propres de `M`. Si les valeurs propres ne sont pas classées par ordre décroissant, il existe les fonctions `fliplr` et `flipud` pour les réordonner.

A l'aide de ces outils, construire le codeur et le décodeur ACP.

Application

- Comparer le signal original au signal après compression/décompression.
- Comparer les distributions de f avant et après compression/décompression.
- Définir une variable raisonnable qui mesure la *compression* du signal lorsque toutes les variables principales ne sont pas transmises au décodeur.
- Etudier l'évolution de p_{mi} en fonction du pourcentage d'*information* transmise et en fonction de la compression du signal.
- Conclusions.

7. ACP : astronomie, analyse de l'ensemble des quasars connus difficulté : simple

Objectifs L'un des principaux buts du Sloan Digital Sky Survey est de trouver les objets les plus lointains jamais observés : les quasars. La lumière émise par ces objets distants a mis des milliards d'années à nous parvenir. Leur observation nous informe donc sur l'histoire primordiale de l'univers. Des données ont été mesurées sur plus de 46 000 quasars, cependant, pour certains, une partie de l'information est manquante.

Vous pouvez lire les données expérimentales dans ces fichiers ascii :

- `quasar_names.txt` = noms des quasars.

- `quasar_variable.txt` = noms des variables.
- `quasar_X` = valeurs de variables pour chaque quasar.

Voici la signification des 22 variables:

- R.A. = Right Ascension.
- Dec. = Declination (Right Ascension & Declination: position dans le ciel en coordonnées équatoriales).
- z = redshift (donne la distance physique et l'âge de l'univers quand la lumière détectée a été émise).
- Radio = amplitude dans la bande des radiofréquences (-1 = pas de donnée, 0 = non détectée).
- X-ray = amplitude dans la bande des rayons X (-9 = pas de données).
- M_i = amplitude absolue dans la bande i.
- x_mag = amplitude (0 = pas de données) à travers le filtre x, correspondant à un domaine donné dans le spectre électromagnétique:
  - g et r pour le spectre visible
  - u pour l'U.V. (émission provenant du disque d'accrétion autour du trou noir central ainsi que des étoiles de la galaxy hôte)
  - i, z pour l'infra-rouge proche (émission de l'anneau de poussière à l'extérieur du disque d'accrétion ainsi que de la formation d'étoiles dans la galaxie hôte)
  - J, H, K pour l'infra-rouge lointain (lumière absorbée puis ré-émise par les poussières interstellaires).

  L'organisation des filtres u, g, r, i, z du SDSS est expliquée ici:
  https://www.sdss3.org/dr9/imaging/imaging_basics.php
  Les domaines électromagnétiques sont donnés là:
  https://en.wikipedia.org/wiki/Photometric_system
- sig_x = barre d'erreur à 1 sigma pour le filtre x.

Analyse des données

- Réaliser l'analyse en composantes principales de la matrice `X`. Faites-le d'abord en utilisant la fonction Matlab `pca` (si la toolbox statistique est disponible, c'est n'est pas toujours le cas) puis en codant vous-même les opérations nécessaires à l'ACP (en Matlab, vous pouvez utiliser la fonction `[E,D]=eig(V)` pour calculer les valeurs/vecteurs propres).
- Tracer les résultats (fonction Matlab `plot`).
- Tirer le maximum de remarques et de conclusions de votre analyse. La fonction Matlab `histogram` (ou `hist`) peut être utile pour visualiser les distributions individuelles.
- Est-ce que la séparation en deux nuages est due à des causes physiques ou instrumentales ? Expliquer.
- Comment peut s'expliquer la position du quasar 150807.25-000940.1 ?

8. ACP : compression du son difficulté : moyenne

Objectifs Le but de ce projet est de construire un codeur et son décodeur utilisant l'ACP, et d'analyser le signal transmis en fonction de différents paramètres. Ici, codage signifie compresser le signal de sorte que sa transmission soit plus rapide et qu'il prenne moins de place en mémoire. L'ACP sera utilisée pour conserver le maximum d'information en utilisant le minimum de variables. Pour pouvoir facilement nous rendre compte des conséquences de la compression du signal, nous travaillerons avec des signaux sonores.

Codage

- La fonction Matlab pour lire un fichier .wav (fichier son) est `[y,Fs] = audioread('file_name.wav')`. Pour écouter l'enregistrement sonore, taper `playblocking(audioplayer(y,Fs))`. Vous pouvez utiliser différents fichiers sons: `bubbles.wav`, `tada.wav`, `speech_8kHz.wav` ("*the discrete Fourier transform of a real-valued signal is conjugate symmetric*").
- Pour comprimer le signal à l'aide de l'ACP, vous devez d'abord le transformer en matrice `X` (voir le cours). Il faut, pour cela, définir un regroupement des canaux. Par exemple, si le regroupement est 10, alors la première colonne de `X` est composée des 1er, 11ème, 21ème ... canaux, la seconde colonne, des 2nd, 12ème, 22ème ... canaux, etc. Pour réaliser le regroupement, vous pouvez utiliser la fonction Matlab `reshape`. Si le nombre de composantes du vecteur `y` n'est pas un multiple de 10 (ou du regroupement), rajouter des 0 à la fin pour qu'il le devienne.
- Pour calculer les vecteurs/valeurs propres d'une matrice, vous pouvez utiliser la fonction `[E,D]=eig(V)`. Si les valeurs propres ne sont pas rangées par valeurs décroissantes, on peut réordonner les matrices à l'aide de `fliplr` et `flipud`.

Utiliser ces outils pour construire le codeur (compression) ACP et son décodeur (décompression).

Application Utiliser le fichier son `speech_8kHz.wav` et répondre aux questions suivantes :

(a) Pour un groupement de 100, quel est le nombre minimum de composantes principales à transmettre de sorte que vous puissiez encore comprendre le message ? quel est le pourcentage d'espace mémoire économisé ?

(b) En ACP, le nombre de lignes de la matrice `X` doit toujours être plus grand que le nombre de colonnes. Quel est le regroupement maximum?

(c) On veut que 90% de l'espace mémoire soit économisé. Complétez le Tableau 1, où la distorsion D est définie par $D = \frac{1}{n_y}\sum_{i=1}^{n_y}(y_i - y_{i\,\text{decode}})^2$, y étant le son original et y_{decode} le son après décodage.

regroupement	10	100	190
D			

Table 1: Distortion pour une économie d'espace mémoire de 90%

Pour une économie de mémoire donnée, vaut-il mieux choisir un regroupement grand ou petit ?

(d) Si vous en avez le temps, démontrez, soit mathématiquement soit par le programme, que $D = \frac{I_{\text{tot}} - I}{regroupement}$, où I_{tot} est l'information totale et I est l'information transmise.

9. ACP : supermarchés difficulté : simple

Objectifs Une grande enseigne souhaite connaître la manière dont ses supermarchés, répartis sur le territoire français, sont perçus par sa clientèle. Un sondage a été effectué demandant aux sondés d'attribuer des notes de 1 à 5 à leur supermarché. Les notes portaient sur la facilité d'accès du supermarché en voiture (Fac), l'étendue du choix proposé (Cho), la disponibilité des vendeurs (Dis), leur compétence (Com) et leur courtoisie (Cou). Pour chaque supermarché, on a retenu la moyenne de ces notes. Elles se trouvent dans le fichier ascii ACP_supermarche.txt. L'ordre des villes est le suivant: Béthune, Le Havre, Rennes, Angers, Nantes, Limoges, Bordeaux, Bayonne, Pau, Toulouse, Nîmes, Valence, Lyon, Dijon, Paris, Reims, Metz, Annecy, Poitier.

Analyse

- Effectuer une ACP de ce tableau à l'aide de la fonction `[E,D]=eig(V)` de Matlab.
- Visualiser les résultats (fonction `plot` de Matlab).
- En tirer le maximum de conclusions.
- Les résultats de la ville d'Amiens vous sont arrivés en retard :

Fac	Cho	Dis	Com	Cou
4.5	1.3	4.0	3.9	4.9

Table 2: Résultats pour le supermarché d'Amiens

Projeter Amiens sur votre plan principal en le traitant comme un individu passif. Conclusions ?

10. ACP : l'économie du secteur des technologies de l'information difficulté : assez élevée

Objectifs Chaque année l'INSEE publie des statistiques relatives à l'activité industrielle et des services en France. Ces données sont accessibles sur internet:

http://www.insee.fr/fr/themes/detail.asp?reg_id=0&ref_id=esa-service-2011

Nous en avons extrait les données relatives au secteur des technologies de l'information :

Les secteurs sont:

(a) `InP` Programmation informatique
(b) `InC` Conseil en systèmes et logiciels informatiques
(c) `InG` Gestion d'installations informatique
(d) `Ina` Autres activités informatiques
(e) `Tfi` Télécommunications filaires
(f) `TSF` Télécommunications sans fil
(g) `Tsa` Télécommunications par satellite
(h) `Taa` Autres activités de télécommunication
(i) `Wth` Traitement de données, hébergement et activités connexes
(j) `Wpi` Portails Internet
(k) `Wap` Activités des agences de presse
(l) `Wau` Autres services d'information n.c.a.
(m) `Roé` Réparation d'ordinateurs et d'équipements périphériques
(n) `Réc` Réparation d'équipements de communication
(o) `Rgp` Réparation de produits électroniques grand public

Chaque secteur est décrit par des grandeurs :

(a) `PGn` Nombre d'entreprises
(b) `PGc` Chiffre d'affaires
(c) `PGs` Effectif salarié moyen
(d) `PGo` Effectif occupé moyen
(e) `PGv` Valeur ajoutée HT
(f) `PGe` EBE
(g) `PGf` Frais de personnel
(h) `PGi` Investissements
(i) `CAe` Entreprises
(j) `CAg` Entreprises du même groupe
(k) `CAh` Entreprises hors groupe
(l) `CAa` Administrations
(m) `CAp` Particuliers
(n) `CAd` Particuliers (services rendus hors domicile)
(o) `CAà` Particuliers (services rendus à domicile)
(p) `CLn` Clientèle nationale
(q) `CLé` Clientèle étrangère
(r) `CLe` Union européenne
(s) `CLh` Hors Union européenne
(t) `STi` Total Sous-traitance incorporée
(u) `STé` Sous-traitance d'études & prestat, de services
(v) `STm` Sous-traitance de matériels & équipements

Le but de ce projet est de tirer le maximum d'informations pertinentes de ces données.

Codage Ecrire un programme effectuant les taches suivantes:

- Lire les données dans le fichier ascii `techno_info.txt` à l'aide de la fonction Matlab `load`.
- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X` (vous n'êtes pas obligés de les utiliser). La transposée de la matrice `M` est donnée par `M'`.
- Pour calculer les valeurs propres et les vecteurs propres de `M`, vous pouvez utiliser la fonction Matlab `[E,D]=eig(M)`. La matrice `E` contient les vecteurs propres (en colonne) et la matrice `D` est diagonale avec pour composantes les valeurs propres de `M`. Les valeurs propres sont parfois classées par ordre croissant. Pour inverser l'ordre, vous pouvez utiliser `fliplr` et `flipud`.
- A l'aide de ces outils, effectuer l'ACP.

Analyses Justifiez chacune de vos réponses.

- La fonction calculant les valeurs propres doit émettre un warning ou un code erreur. Ceci est dû au fait que pour que l'APC soit possible, il faut que le tableau individus-caractères contienne plus d'individus que de caractères. Trouver quelles sont les variables inutiles car combinaisons linéaires d'autres variables.
- Supprimer le nombre suffisant d'autres variables (celles qui vous semblent les moins intéressantes) de sorte que la matrice `X` devienne verticale.
- La matrice `X` contient des -1 qui correspondent à l'absence de données. Pour éviter de supprimer des variables ou des individus lorsqu'ils sont incomplets, on remplace souvent les valeurs manquantes par des valeurs "raisonnables", souvent des moyennes calculées à partir des autres individus et des autres variables. Remplacez les -1 par des valeurs qui vous semblent raisonnables
- Certaines variables sont très corrélées pour des raisons triviales (comme les frais de personnel et les effectifs salariés moyens). Cherchez si des transformations de ces variables ne pourrait pas apporter des informations nouvelles permettant de caractériser les secteurs entre eux.
- Analysez les résultats des ACP que vous aurez obtenus avec différentes versions de la matrice `X` et tirez-en des conclusions sur l'économie du secteur des technologies de l'information.

11. ACP : ouverture mondiale de l'économie en 2017 difficulté : simple

Objectifs Le but de ce projet est d'étudier le taux d'ouverture mondiale de l'économie à partir des chiffres fournis pas l'INSEE en 2017. Le dossier de 12 pages de l'INSEE se trouve dans le folder `students_etudiants`, fichier `ouverture_mondiale.pdf`, les données dans `ouverture_mondiale.txt`. Il n'est pas nécessaire de lire entièrement ce rapport, mais vous y trouverez les informations sur la signification des variables utilisées. Le tableau de données se trouve en dernière page.

L'objectif est de tirer le maximum d'informations pertinentes de ces données.

ACP

- Pour calculer les valeurs propres et les vecteurs propres de `M`, vous pouvez utiliser la fonction Matlab `[E,D]=eig(M)`. La matrice `E` contient les vecteurs propres (en colonne) et la matrice `D` est diagonale avec pour composantes les valeurs propres de `M`. Si les valeurs propres ne sont pas classées par ordre décroissant, vous pouvez utiliser `fliplr` et `flipud`, pour réordonner les matrices.
- A l'aide de ces outils, effectuer l'ACP.

Analyses Analyser les corrélations entre les variables. Que peut-on déduire du nuage des pays ? etc.

12. ACP : compression des photos prises sur Mars difficulté: prolongement du TP codage

Objectifs Le but de ce projet est de construire le codeur ACP du rover Curiosity pour lui permettre de comprimer de manière optimum ses photos de la surface de Mars avant de les envoyer vers la Terre. Le décodeur correspondant, utilisé par le NASA Space Center sur Terre, sera également écrit.

Les images `Mars_dunes.jpg` et `Mars_Path_Finder.jpg` du répertoire `students_etudiants` ont été téléchargées du site de la NASA:

http://mars.nasa.gov/msl/multimedia/images/?ImageID=7539

L'ACP a été utilisée pour condenser le maximum d'information en utilisant le minimum de variables. Initialement, l'image est composée de trois matrices, une pour chaque couleur de base. Chaque composante correspond à un pixel.

La procédure utilisée par Curiosity est fondée sur le même principe, mais comprend des raffinements supplémentaires.

Codage

- Pour charger un fichier .jpeg et visualiser l'image, taper:

```
Yini  = single(imread('Mars_dunes.jpg'));
ltot  = size(Yini,1);
ctot  = size(Yini,2);
trois = size(Yini,3);
image(uint8(Yini))
title('image initiale')
axis equal
```

- L'image n'est pas comprimée de manière globale mais bloc par bloc. Un bloc `Y` est un morceau rectangulaire de `l` lignes par `c` colonnes de la matrice `Yini`. Il vaut mieux d'abord choisir le nombre de blocs que l'on souhaite en horizontal et en vertical et écrire :

```
nl = 5;
l  = floor(ltot/nl);
```

  Chaque bloc `Y` doit être extrait de la matrice `Yini` et transformée en matrice `X` dans laquelle les lignes `i` sont les pixels du bloc et les colonnes `j` les trois composantes de la couleur du pixel. Cela peut être effectué à l'aide de la fonction Matlab `reshape`.

- Ecrire le codeur ACP dans une fonction indépendante:

```
function [P,E,Ip] = codeur_ACP(X,p)
```

  où `p` is the nombre de variables principales que Curiosity doit transmettre à la Terre.

  `P`, matrice des variables principales à `p` colonnes

  `E`, matrice correspondante des vecteurs propres

  `Ip`, vecteur contenant les pourcentages d'information portée par chaque variable principale.

- Ecrire le décodeur ACP dans une fonction indépendante:

```
function X = decodeur_ACP(P,E)
```

- Utiliser trois boucles imbriquées sur `p`, `i` et `j`, où `i` et `j` sont les numéros des blocs le long des lignes et des colonnes. Pour chaque bloc, construire la matrice `X`, la coder puis la décoder. Transformer la matrice décodée en une matrice `Y` (`l` par `c` par 3). Enfin, insérer le bloc `Y` au bon endroit dans la matrice `Yfin` finale (de même taille que `Yini`).

- Visualiser la matrice `Yfin` pour chaque valeur de `p`.

Analyses

- Pour chaque `p`, calculer le pourcentage moyen (sur les blocs) de l'information ACP que Curiosity a envoyé vers la Terre.
- Pour chaque `p`, calculer, à l'aide de `numel`, une quantité proportionnelle au nombre de bits envoyés vers la Terre.
- Commenter les résultats. Vous pourriez suggérer une recette pour déterminer les valeurs optimum des différents paramètres du problème, ou procéder à d'autres analyses ...
- Si vous en avez le temps, appliquez la même procédure mais sans réaliser d'ACP. Curiosity envoie les `p` premières colonnes de la matrice `X` ainsi que les valeurs moyennes des `3-p` dernières colonnes. Le décodage consiste à remplacer les `3-p` dernières colonnes par leur valeur moyenne. Faire les mêmes analyses que dans le cas de l'ACP, commenter, conclure.

voir l'exemple d'image qui n'a pas pu être correctement décodée à cause de problèmes de transmission, Fig. 2.

Figure 2: Exemple d'image avec des blocs à différents niveaux de décodage.

13. **ACP : sujet libre** difficulté : simple à élevée selon vos données et la richesse de votre analyse

Objectifs Choisissez un ou des tableaux de données que vous souhaitez analyser à l'aide de l'ACP.

Codage

- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X`.
- La transposée de la matrice `M` est donnée par `M'`.
- La fonction Matlab `[E,D]=eig(M)` calcule les vecteurs/valeurs propres de `M`. Faites attention au fait que les valeurs propres sont parfois classées par ordre croissant. Pour inverser l'ordre, il existe les fonctions `fliplr` et `fliplr`.

A l'aide de ces outils, construire le code ACP.

Application Tirer le maximum de conclusions du (des) tableau(x) de données étudié(s).

14. AFC : les prix Nobel difficulté : simple

Objectifs Le but de ce projet est d'étudier les corrélations entre catégorie de prix Nobel et origine géographique des lauréats. Les données datent de 2015, elles viennent du site suivant :
http://www.nobelprize.org/nobel_prizes/lists/all/index.html
Elles se trouvent dans les fichiers `Nobel_N.txt`, `Nobel_disciplines.txt` et `Nobel_sous-continents.txt` du répertoire `students_etudiants`.

Analyse

- La fonction `load` permet de lire le fichier de données. Effectuer une AFC du tableau (éventuellement à l'aide de la fonction `eig` de Matlab).
- Visualiser les résultats
- En tirer le maximum de conclusions.

15. AFC : analyse d'un profil de clientèle difficulté : simple

Objectifs Le but de ce projet est d'étudier le profil de la clientèle d'un supermarché. Chaque article est décrit par la classe de produits auquel il appartient : **Hyg**iène du corps, **Ent**retient de la maison, **Leg**umes, **Boi**ssons, **Via**ndes et poissons, **Fru**its, autres **Nou**rritures, **Vet**ements, **Jar**din, **Cul**ture, **Bla**nc, **Aut**omobile, **Bru**n. Sur un an, les couples volontaires (anonymes) ont accepté que le supermarché note qui avait effectué chacune des visites (passage en caisse) : l'homme seul, la femme seule ou les deux ensemble. Les produits achetés sont classés de la manière suivante :

- Si le produit est acheté à plus de 50% des passages en caisse par le couple ensemble, il est compté dans la classe **ens**emble.
- Si le produit est acheté à plus de 50% des passages en caisse par la femme seule, il est compté dans la classe **fem**me.
- Si le produit est acheté à plus de 50% des passages en caisse par l'homme seul, il est compté dans la classe **hom**me.
- Sinon, il est compté dans la classe **alt**ernativement.

Les données se trouvent dans le fichier `profils_clients.txt` du répertoire `students_etudiants`. Les classes en ligne et en colonne sont ordonnées comme dans le texte ci-dessus.

Analyse

- La fonction `load` permet de lire le fichier de données. Effectuer une AFC du tableau (éventuellement à l'aide de la fonction `eig` de Matlab).
- Visualiser les résultats
- En tirer le maximum de conclusions.

Analyse

- La fonction `load` permet de lire le fichier de données. Effectuer une AFC du tableau (éventuellement à l'aide de la fonction `eig` de Matlab).
- Visualiser les résultats
- En tirer le maximum de conclusions.

16. **AFD : sujet libre** difficulté : simple à élevée selon vos données et la richesse de votre analyse

Objectifs Choisissez un ou des tableaux de données que vous souhaitez analyser à l'aide de l'AFC.

Codage

- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X`.
- La transposée de la matrice `M` est donnée par `M'`.
- La fonction Matlab `[E,D]=eig(M)` calcule les vecteurs/valeurs propres de `M`. Faites attention au fait que les valeurs propres sont parfois classées par ordre croissant. Pour inverser l'ordre, il existe les fonctions `fliplr` et `fliplr`.

A l'aide de ces outils, construire le code AFC.

Application Tirer le maximum de conclusions du (des) tableau(x) de données étudié(s).

17. **ACP AFC AFD : fun with flags** difficulté : simple à élevée

Objectifs A l'aide des techniques d'analyse multi-dimensionnnelles analysez les corrélations pouvant exister entre les caractéristiques d'un pays et les couleurs de son drapeau.

Commencez avec un nombre réduit de pays que vous pourrez compléter par la suite. Vous pouvez éventuellement apporter d'autres données relatives aux pays pour les comparer avec les couleurs.

18. **AFD : sujet libre** difficulté : simple à élevée selon vos données et la richesse de votre analyse

Objectifs Choisissez un ou des tableaux de données que vous souhaitez analyser à l'aide de l'AFD.

Codage A l'aide des outils suivants, construire le code AFD.

- Les fonctions Matlab `mean`, `std` et `repmat` peuvent être utiles pour standardiser la matrice `X`.
- La transposée de la matrice `M` est donnée par `M'`.
- La fonction Matlab `[E,D]=eig(M)` calcule les vecteurs/valeurs propres de `M`. Faites attention au fait que les valeurs propres sont parfois classées par ordre croissant. Pour inverser l'ordre, il existe les fonctions `fliplr` et `fliplr`.

Application Tirer le maximum de conclusions du (des) tableau(x) de données étudié(s).

19. AFD : caractéristiques de molécules obtenues par modèle ab-initio à l'aide de différentes méthodes et bases difficulté : plus difficile

Objectifs Une nouvelle méthode de dépollution des hydrocarbures atmosphériques, développée dans les laboratoires de la faculté des sciences d'Orsay, consiste à fragmenter les grosses molécules (le plus polluantes) à l'aide d'un plasma froid. Pour prévoir et optimiser le fonctionnement des plasma de dépollution, il est nécessaire de connaître les caractéristiques physiques des molécules polluantes et de leurs fragments. Les caractéristiques des 66 molécules produites dans la fragmentation du propène (CH_3-CH_2-CH_3) ont été calculées à l'aide d'un code quantique ab-initio (code Gaussian). Ces caractéristiques sont:

- E_f l'énergie fondamentale de la molecule
- frq la moyenne géométrique des fréquences de la molécule ($frq = (\prod_{j=1}^{n} \nu_j)^{1/n}$)
- I_x les trois moments d'inertie
- I_y
- I_z
- R le rayon moyen de la molécule
- $E_{\mathrm{d\,min}}$ son énergie de dissociation minimum

Dans ce code, il est possible de choisir quelles méthode théorique de calcul est utilisée ainsi que la base sur laquelle les fonctions d'onde sont décomposées. Sept jeux de données ont été générés pour les mêmes molécules pour différents choix de méthode/base:

- G* : méthode DFT , base b3lyp, avec fonctions de diffusion
- G** : méthode DFT , base b3lyp, avec fonctions de diffusion et de polarisation
- CCSD(T)Dζ : méthode Coupled-Cluster, base comprenant une double fonction ζ
- CCSD(T)Tζ : méthode Coupled-Cluster, base comprenant une triple fonction ζ
- CCSD(T)Qζ : méthode Coupled-Cluster, base comprenant une quadruple fonction ζ
- CCSD(T)5ζ : méthode Coupled-Cluster, base comprenant une quintuple fonction ζ
- G2 : méthode composite.

Pour les jeux CCSD, G** a été utilisé pour calculer les caractéristiques géométriques (frq, I_x, Iy, Iz, R_fr), la méthode coupled-clusters est utilisée pour les énergies uniquement (E_gs, $E_{\mathrm{d\,min}}$). Le temps CPU pour les jeux CCSD vont de la minute par molécule (Dζ) à 1 à 3 jours par molécule (5ζ). Le temps CPU pour les jeux DFT sont respectivement de l'ordre de la minute et de l'heure par molécule. La méthode composite est réputée donner des résultats précis pour des temps CPU relativement courts (1 à 3 heures par molécule).

Notre but est de montrer les différences entre les jeux, trouver lesquels sont similaires, quelles variables séparent les jeux, etc. Nous voulons aussi vérifier si G2 semble donner des résultats fiables.

Analyses

- Les données se trouvent dans le fichier `ab_initio_X.txt`. La matrice contient 7×66 molécules (les molécules d'un jeu donné sont consécutives) en ligne et 7 variables en colonne. les fichiers `ab_initio_variables.txt`, `ab_initio_bases.txt`, `ab_initio_molecules.txt` contiennent respectivement le nom des variables, les méthodes/bases et les molécules (c signifie que les 3 carbones forment un cycle, t signifie que la molécule est dans son état triplet).
- Effectuer une AFD de ces données et en tirer le maximum de conclusions.

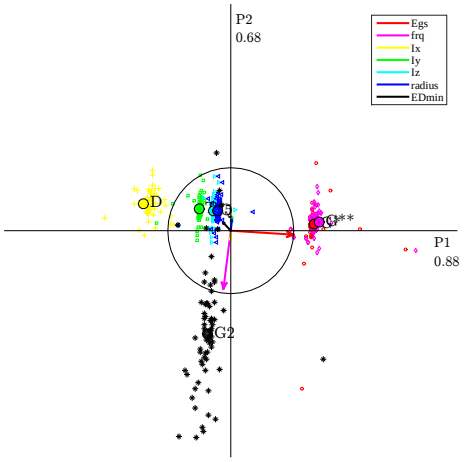

Figure 3: Plan discriminant

- Le physicien retrouve dans son ordinateur un fichier avec les caractéristiques suivantes : $E_{\mathrm{f}} = -3.6\,10^{-4}$, $frq = 3.6\,10^{-1}$, $I_x = 4.2$, $I_y = 5.7\,10^{-3}$, $I_z = 7.9\,10^{-3}$, $R = -9.7\,10^{-3}$, $E_{\mathrm{d\,min}} = -9.4\,10^{-3}$. Pouvez-vous l'aider à déterminer avec quelles méthodes/bases cette molécule à été calculée ?

Vous devriez trouver le résultat de la figure 4

Chaînes de Markov

1. Entreprise de confection artisanale

Difficulté : facile

Une entreprise de confection souhaite analyser son processus de fabrication et de vente pour évaluer ses coûts de revient. La chaîne représente le processus commercialisation/fabrication d'une veste sur mesure. Il commence par la prise de commande avec une première prise des mesures du client par le tailleur (état Com). Cela déclenche les travaux de couture en atelier (état Ate) avec la probabilité 1. Suite à ces travaux, la veste peut soit partir pour essayage par le client en magasin (état Ess) avec une probabilité 0,8 soit nécessiter de nouvelles prises de mesures par le tailleur (état Mes) avec une probabilité 0,2. Après les nouvelles prises de mesure, la veste retourne toujours en atelier. Suite à l'essayage en magasin, la veste est soit encaissée (état Enc) avec une probabilité 0,8 soit envoyée pour de nouvelles mesures par le tailleur (état Mes) avec une probabilité 0,2.

Les coûts de reviens de chaque opération sont les suivants : Com = 100€, Ate = 300€, Ess = 50€, Mes = 100€, Enc = 20€.

Donnez le maximum de propriétés de ce processus en utilisant ce qui a été vu en cours. Calculez le coût de revient moyen d'une veste et la distribution des prix possibles (différentes techniques de calcul étant possibles, si vous en imaginez plusieurs indiquez les toutes sur le compte-rendu).

2. Gestion des eaux pluviales de la ville de Hanoi

Difficulté : moyenne

Vous travaillez pour l'entreprise Veolia qui gère les eaux de la ville de Hanoi.

Voici une représentation simplifiée du système d'évacuation des eaux pluviales à Hanoi. Dans la suite, tous les taux sont donnés en heures^{-1}. L'eau arrive sur Hanoi apportée par les nuages (état N). L'eau peut tomber sur des surfaces imperméables (état I) telles que des routes, des parkings ou des bâtiments avec un taux $i = 10$, dans des lacs (état L, taux $l = 1$) et sur les surfaces perméables (état P, taux $p = 2$). A partir des surfaces imperméables, l'eau s'écoule vers les surfaces perméables (taux $q = 1{,}5$) ou, à travers les égouts, vers le fleuve rouge (état R, taux $v = 1{,}7$). L'eau dans les lacs coule vers le fleuve avec le taux $r = 3$. L'eau dans le fleuve rouge s'évacue vers la mer (état M, taux $m = 14$). L'eau de pluie tombée sur les surfaces perméables coule vers le fleuve avec le taux $w = 0{,}1$ ou est absorbée par le sous-sol (état S, taux $s = 18$). Les eaux du sous-sol rejoindront la mer (taux $a = 9$) ou seront extraites par des pompes familiales pour être bues directement par les habitants (état H, taux $y = 0{,}01$) ou seront extraites par la station de pompage de la ville (état X, taux $x = 0{,}1$) pour être distribuées, après traitement, à la population (taux $z = 0{,}3$).

L'objectif de cette analyses est d'extraire de le maximum d'information de ce processus de Markov. Les mêmes résultats peuvent souvent être obtenus de différentes manières, indiquer toutes les méthodes utilisées.

Deux questions spécifiques:

- En ajoutant des états à la chaîne, trouvez une méthode pour déterminer la proportion d'eau non traitée absorbée par les habitants.
- En tant que gestionnaire de ce réseau de collecte et de distribution d'eaux, sur quels paramètres pouvez-vous jouer pour diminuer la quantité de pollution absorbée par la population buvant de l'eau non traitée?

3. Dépollution

Difficulté : moyenne

Le propane C_3H_8 est un gaz polluant émis durant la combustion. Pour détruire ces molécules, émise par les pot d'échappement des automobiles ou par les cheminées d'usines, on peut utiliser un plasma froid. The flux de molécules de propane traverse le plasma où elles sont excitées par les électrons libres et les ions. Les molécules sont alors cassées en fragments. Chaque ensemble de fragments d'un propane est appelé une *partition*. Le but est de réduire le propane en molécules plus petites moins polluantes. La fragmentation du propane dans le plasma utilisé conduit à 9 partitions possibles (voir la figure 4). Chaque flèche correspond à la cassure d'une molécule en deux plus petites. Les valeurs sur les flèches sont appelées "rapports de branchement" est sont équivalentes à des taux de transition.

Les valeurs des rapports de branchement (en ms^{-1}) sont les suivantes : $a = 13, b = 23, c = 109, d = 235, e = 456, f = 53, g = 58, h = 1242, i = 214, j = 733, k = 2355, l = 1754, m = 514$. Combien de temps les molécules doivent-elle rester dans le plasma pour que la moitié des propanes soient réduits dans la neuvième partition (la moins polluante) ? Si on suppose que la fragmentation s'arrête aux partitions 7 et 8, quelle seraient les proportions relatives de CH_3 et de C_2H_4 dans le gaz final après un temps très long dans le plasma ?

4. Gestion de crise : protection de la population des retombées de Fukushima

Difficulté : moyenne

Vous travaillez pour la préfecture de la région de Fukushima qui vous demande d'étudier la propagation du césium radioactif et la contamination de la population.

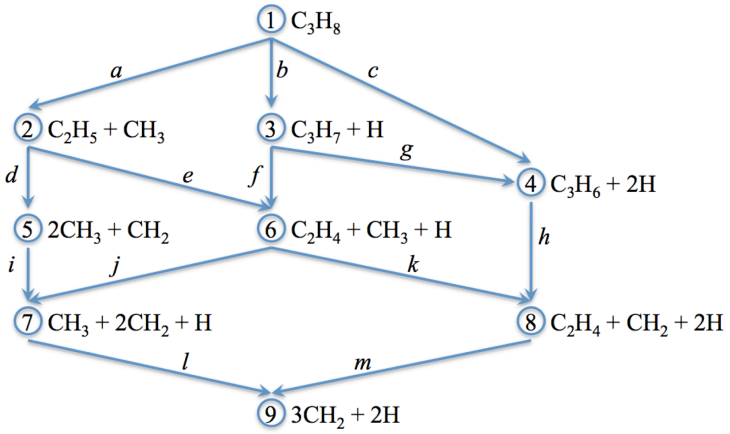

Figure 4: Fragmentation séquentielle du propane.

Suite aux explosions dans la centrale de Fukushima, du césium radioactif ^{137}Cs a été relâché dans l'atmosphère (état A). Une partie du césium est retombée en mer (état M) avec le taux de transition $m = 1/2$ jour^{-1} où il a pu être absorbé par des poissons (état P, taux de transition $p = 1/3$ mois^{-1}) qui peuvent rejeter le césium dans l'eau (taux de transition $r = 1/10$ jour^{-1}). D'autres atomes de césium tombent sur la terre (état T, taux de transition $t = 1/15$ jour^{-1}). Ils pourront alors être absorbés par des plantes comestibles (états C, taux de transition $c = 1/2$ jour^{-1}). Le césium tombé sur la terre et celui absorbé par les plantes peuvent diffuser dans le sous-sol (état S) avec le même taux de transition $s = 1$ mois^{-1}. Les humains (états H) peuvent absorber le césium en mangeant du poisson (taux de transition $x = 1$ mois^{-1}), des fruits et légumes (taux de transition $z = 1$ mois^{-1}) et en inhalant de l'air (taux de transition $i = 1/4$ mois^{-1}). Le césium dans le corps humain peut être éliminé avec les urines (état E, taux de transition $e = 1/10$ an^{-1}). N'oubliez pas de tenir compte du fait que le césium est radioactif, il se transforme en baryum.

Analyser le plus complètement possible cette chaîne de Markov grâce à son évolution temporelle et à l'aide des méthodes vues en cours.

Les autorités veulent savoir s'il faut en priorité interdire aux habitants de la préfecture de manger du poisson ou des plantes. Une méthode, parmi d'autres, pour trouver ce résultat est d'écrire l'équation différentielle qui donne $\mathrm{d}H_\mathrm{P}$ et $\mathrm{d}H_\mathrm{C}$, les quantités de césium absorbées par les humains entre les instants t et $t+\mathrm{d}t$ en mangeant du poisson et des fruits et légumes. Dans votre programme Matlab, vous pouvez sommer ces petites quantités au cours du temps pour obtenir les quantités totales.

Une autre méthode consiste à modifier la chaîne de Markov pour qu'elle donne directement le résultat sous forme d'une matrice d'absorption $\mathbf{A}$.

5. Matrice de Google appliquée à la gestion des affaires

Difficulté : plus difficile

L'organisation d'une vraie entreprise de consulting a été modélisée à l'aide d'un graphe orienté constitué de noeuds représentant les entités de l'entreprise, reliés par des liens représentant les relations de cause à effet. Ce graphe peut être vu comme une chaîne de Markov. La matrice de Google a été appliquée au graphe organisationnel de l'entreprise pour étudier l'importance de ses différentes entités. Les résultats de cette études sont parus dans l'article suivant :

"Google matrix of business process management" par M.W. Abel et D.L. Shepelyansky (fichier `article business management.pdf` dans le répertoire `students_etudiants`, les parties barrées ne sont pas indispensables pour ce projet sauf si vous le terminez en avance).

Les données sont accessibles à l'adresse suivante :

http://www.quantware.ups-tlse.fr/QWLIB/cheirankbusiness/

Elles ont été recopiées dans les fichiers `business_noeuds.txt` et `business_liens.txt` dans le répertoire `students_etudiants`.

Vérifiez de manière aussi complète que possible les résultats présentés dans l'article. Testez ensuite vos propres idées pour caractériser cette matrice de Google et cette entreprise. Vous pouvez, par exemple, étudier la convergence du vecteur population vers la population à l'équilibre. Donnez vos conclusions sur ce que nous apprend cette étude sur l'entreprise de consulting.

6. Simulation du moteur de recherche de Google

Difficulté : plus difficile

Ecrivez votre propre moteur de recherche Google. Il devra trouver, dans un ensemble de pages fictives que vous aurez créée, les pages contenant les mots demandés par l'utilisateur puis présenter le résultat en classant les pages par ordre décroissant de PageRank.

(a) Commencez par créer des pages fictives qui peuvent, par exemple, être des fichiers .txt contenant du texte et des pointeurs vers d'autres pages fictives. Ces pointeurs peuvent être simplement le nom du fichier .txt correspondant à la page pointée, précédés du mot "pointeurvers:"

(b) Attribuer un PageRank à chacune des pages grâce à la technique de la matrice Google

(c) Ecrivez un moteur de recherche qui ira chercher les pages contenant les mots demandés (commencer par un mot). La liste des pages trouvées devra être présentée à l'utilisateur par ordre décroissant de PageRank.

Vous débuterez peut-être ce projet avant que le cours sur la matrice Google n'ai eu lieu. La matrice Google contient les probabilités de transition des "surfers" d'une page internet à une autre. Vous pouvez débuter le projet en créant une matrice bidon (ne contenant pas d'état absorbant) et en écrivant un programme qui calcule l'évolution temporelle du vecteur population et la visualise. Après le cours sur Google, vous remplacerez votre matrice par une matrice Google.

7. Le PageRank de Google

Difficulté : moyenne

Le but de ce projet est d'analyser et de caractériser le plus complètement possible l'algorithm PageRank de classement des pages internets utilisé par Google. Vous pouvez utiliser toutes les informations disponibles sur internet. Il convient au minimum d'écrire un code permettant de tester cet algorithm et d'observer l'influence des différents paramètres. Ce code devra permettre de traiter un réseau contenant un très grand nombre de pages, chacune comportant un petit nombre (éventuellement 0) de liens. Vous rendrez un rapport aussi complet et original que possible.

Vous débuterez probablement ce projet avant que le cours sur la matrice Google n'ai eu lieu. La matrice Google contient les probabilités de transition des "surfers" d'une page internet à une autre. Vous pouvez débuter le projet en créant une matrice bidon (ne contenant pas d'état absorbant) et en écrivant un programme qui calcule l'évolution temporelle du vecteur population et la visualise. Après le cours sur Google, vous remplacerez votre matrice par une matrice Google.

8. Rejet de métaux lourds dans la mer

Difficulté : facile

Vous êtes le responsable de la surveillance informatique d'une aciérie. Un bug dans un protocole provoque le rejet de métaux lourds dans l'eau de la mer (état E). Ces atomes peuvent être fixés par le plancton (état Pl, taux de transition $a = 1/1{,}1$ jour^{-1}), le krill (état K, taux de transition $b = 1/5{,}2$ jour^{-1}) et les poissons (état Po, taux de transition $c = 1/6{,}0$ jour^{-1}). Le plancton, le krill et les poissons peuvent relâcher les métaux dans l'eau avec les taux de transition respectifs $d = 1/5{,}2$ jour^{-1}, $e = 1/4{,}1$ jour^{-1} et $f = 1/0{,}5$ jour^{-1}. Les trois finissent par mourir (état M) avec les taux de transition respectifs $i = 1/5$ jour^{-1}, $j = 1/10$ jour^{-1} et $k = 1/2$ année^{-1}. Le krill absorbe également des métaux lourds en mangeant le plancton contaminé (taux de transition $g = 1/6$ jour^{-1}) et les poissons en mangeant du krill (taux de transition $h = 1/2$ jour^{-1}). On fait l'hypothèse que les métaux lourds ont tous été rejetés brutalement lors d'un accident.

La masse totale de plancton est 4 fois supérieure à celle du krill qui est 6 fois supérieure à celle des poissons. Etudier leur contaminations respectives (comment désigne t on ce phénomène) ?

Si vous avez le temps, analysez ce qui se passe si la fuite est continue. Pour cela on ajoute un état A (aciérie) avec un taux de transition très faible vers l'eau de mer ($w = 1/25$ année^{-1}). Il est conseillé de tracer les courbes d'évolution en axes log-log.

9. Le tennis

Difficulté : plus long à coder

Construire la chaîne de Markov d'un "jeu" au tennis. Les états sont tous les scores possibles (0-0, 15-0, 0-15, ..., égalité, avantage A, avantage B). Les joueurs sont caractérisés par leurs probabilités p et $1-p$ de remporter un échange. Est-ce que le jeu amplifie ou diminue les différences de niveau entre les joueurs ? Caractérisez, autant que vous le pouvez, un jeu (et un set et un match si vous avez le temps).

10. Etude du vieillissement des couvertures de toit

Difficulté : simple, lecture de 4 pages en anglais

L'article "Discrete stochastic model for performance prediction of roofing systems" (Modèle aléatoire discret pour la prédiction des performances des systèmes de couverture) par Z. Lounis, M. Lacasse et D. Vania (fichier `toiture.pdf` dans le répertoire `students_etudiants`), propose aux entreprises de couverture une méthode pour calculer le vieillissement moyen des leurs produits. La méthode est décrite dans la partie **4 Markov chain modeling of roofing system performance** (il n'est pas utile d'aller lire les articles et ouvrages donnés en référence). Dans un premier temps, ne lisez que la partie encadrée de l'article, vous ne lirez le reste si vous avez le temps.

Etude des notations de l'article

- D'après le texte, quelle est la relation entre les matrices $\mathbf{P}$, $\mathbf{P}(n)$ et $\mathbf{P}_0$ de l'article et les matrices $\mathbf{M}$ et $\mathbf{n}$ du cours ?
- Ecrire l'équation (3b) avec les notations du cours.
- En une quinzaine de lignes, résumer la méthode développée par les auteurs .

Réalisation

- Programmer la chaîne de la **Figure 1** pour des probabilités que vous choisirez.
- Tracer la courbe de la **Figure 2**.
- Etudier l'influence des probabilités sur la forme de la courbe.
- Comment peut-on obtenir la valeur du temps moyen avant réparation (état 1) sans calculer l'évolution temporelle ?

11. Gestion de 2 boulangeries

Difficulté : facile

Une entreprise de boulange possède deux points de vente P_1 et P_2 dans une ville. Dans le premier, les pains se vendent avec une constant de temps $v_1 = 1/3$ heure^{-1} et dans le second avec une constant de temps $v_2 = 1/5$ heures^{-1}. Une fois vendu, le pain se trouve dans l'état V. Le pain peut devenir impropre à la vente (rassi, moisi, sali etc.) et donc se retrouver dans l'état invendu I au bout d'un temps moyen $T = 10$ heures. Le gestionnaire de l'entreprise peut décider de transférer des pains d'une boulangerie à l'autre. On note $d_1 = 1/60$ minute^{-1} la constante de temps pour le transfert des pains du point de vente P_1 au point de vente P_2 et $d_2 = 0,2$ heure^{-1} la constante de temps de transfert dans l'autre sens.

- Tracer l'évolution temporelle du nombre de pains dans chacun des états.
- En début de journée, une proportion α des pains a été livrée dans le premier point de vente et une proportion $1 - \alpha$ dans le second point de vente. Montrer, grace à votre simulation de l'évolution temporelle et grace au calcul matriciel, que la proportion globale de pains invendus est $p = \frac{i}{\Delta}\left(d_1 + d_2 + (1-\alpha)\, v_1 + \alpha\, v_2 + i\right)$, avec $\Delta = [(v_1 + i + d_1)\,(v_2 + i + d_2) - d_1\, d_2]$.
- Comment l'entreprise doit elle répartir les pains en début de journée pour que la proportion d'invendus soit minimum (expliquer pourquoi ce résultat est logique) ?
- A l'aide des propriétés des chaînes de Markov vues en cours, donnez le maximum de caractéristiques de ce processus.

12. Sujet libre, processus absorbant

Difficulté : simple à difficile

Inventez par vous même une chaîne de Markov absorbante modélisant de préférence, mais pas forcément, un processus relatif à la gestion des entreprises. Il n'est pas nécessaire que vous connaissiez les probabilités de transition, vous pouvez inventer des valeurs. Votre modélisation ne sera pas forcément réaliste, mais vous devrez expliquer en quoi elle l'est et ne l'est pas. Analysez ensuite, de la manière la plus détaillée possible, les caractéristiques de votre chaîne à l'aide des propriétés vues en cours.

13. Sujet libre, processus régulier

Difficulté : simple à difficile

Inventez par vous même une chaîne de Markov régulière modélisant de préférence, mais pas forcément, un processus relatif à la gestion des entreprises. Il n'est pas nécessaire que vous connaissiez les probabilités de transition, vous pouvez inventer des valeurs. Votre modélisation ne sera pas forcément réaliste, mais vous devrez expliquer en quoi elle l'est et ne l'est pas. Analysez ensuite, de la manière la plus détaillée possible, les caractéristiques de votre chaîne à l'aide des propriétés vues en cours.

14. Sujet libre, matrice de Google

Difficulté : simple à difficile

Internet forme ce qu'on appelle un "réseau orienté", c'est-à-dire un ensemble des "noeuds" (les pages) connectés par des "liaisons" (les liens pointant vers d'autres pages). Les liaisons sont "orientées" puisque le lien pointe d'une page vers une autre.

Trouvez sur internet, ou inventez par vous-même, un autre type de réseau orienté (par exemple un ensemble d'entreprises ayant des relations fournisseur-client). Construisez la matrice Google de ce réseau. Tirez-en le maximum d'information à la fois sur le réseau étudié et sur la méthode d'analyse.

Vous débuterez peut-être ce projet avant que le cours sur la matrice Google n'ai eu lieu. La matrice Google contient les probabilités de transition des "surfers" d'une page internet à une autre. Vous pouvez débuter le projet en créant une matrice bidon (ne contenant pas d'état absorbant) et en écrivant un programme qui calcule l'évolution temporelle du vecteur population et la visualise. Après le cours sur Google, vous remplacerez votre matrice par une matrice Google.

Figure 5: L'affiche du film

15. **Le Bon, la Brute et le Truand**

Difficulté : difficile

Trois pistoleros décident de résoudre leur différend par un "duel" à trois. Le premier, le Bon (the **G**ood) a une probabilité $g = 1/3$ de tuer sa cible, la Brute (the **B**ad) tue sa cible avec la probabilité $b = 1/2$ et le Truand (the **U**gly) avec la probabilité $u = 1$. Ils tirent une seule balle, l'un après l'autre, toujours dans le même ordre : G (Bon), B (Brute), U (Truand). Le gagnant est le dernier encore en vie. Le premier objectif est de déterminer lequel des trois a la plus forte probabilité de gagner. On peut faire l'hypothèse évidente que l'intérêt de chaque pistolero est de tirer sur le plus fort des deux autres.

On introduit 10 états: Gbu, gBu, gbU, Gu, gU, Gb, gB, G, B, U qui indiquent les pistoleros encore vivants (la majuscule correspond au tireur). L'état d'entrée est donc Gbu. Tracer la chaîne de Markov correspondante.

Faites une liste des variables possibles (probabilités et temps) qui peuvent caractériser le problème, puis essayez de trouver un nombre maximum de ces valeurs en utilisant l'algèbre matricielle de Markov et la simulation.

(Beaucoup de choses peuvent être étudiées avec 3 pistoleros, il ne serait pas sage d'en envisager plus. Si vous voulez prouver que l'intérêt de chaque tireur est de tirer sur le plus fort des deux autres, je vous recommande d'introduire g_u la probabilité que le Bon choisisse de tirer sur le Truand, b_u la probabilité que la Brute choisisse de tirer sur le Truand et u_b la probabilité que le Truand choisisse de tirer sur la Brute, puis de construire la nouvelle chaîne de Markov et d'en déduire les valeurs optimales des probabilités g_u, b_u et u_b pour chaque tireur).

Intelligence artificielle ces sujets sont plus difficiles

1. **IA : Algorithme MinMax généraliste**

L'objectif ambitieux de ce projet est de construire un programme d'IA, utilisant l'algorithme MinMax, permettant de résoudre n'importe quel jeu (ou problème) à somme nulle et à deux joueurs (ou intervenants). Le code pourra être testé grâce à des routines en Matlab, pour le jeu du morpion.

Le projet consiste à écrire deux routines :

- la fonction `[note, coup] = minmax(position)` où `position` est un enregistrement décrivant la position du jeu (`minmax` n'a pas besoin de connaître la structure de cet enregistrement), `coup` est le coup obtenant la meilleure note (sa structure est la même que celle de `position`) et `note` est la note attribuée à `coup`. Pour simplifier grandement la programmation, il faut que `minmax` soit une fonction *récursive*, c'est-à-dire une fonction qui peut s'appeler elle-même (voir Wikipedia et la doc. de Matlab). Dans ce cas, le code ne devrait comporter qu'une vingtaine de lignes.
- le programme principal pour le jeu de morpion, chargé de l'interface avec l'utilisateur et des appels à `minmax(position)`.

Pour fonctionner pour un jeu ou un problème particulier, votre code fera appel aux fonctions suivantes :

- `liste_coups = fournir_coups(position)`, où `liste_coups` est un vecteur d'enregistrements de même structure que `position` et que `coup`, qui contient tous les coups possibles.
- `note = fournir_note(position)`, où `note` est la note d'une position.
- `afficher_position(position)`, affichage à l'écran.

Ces fonctions, pour le jeu de morpion, se trouvent dans le folder `students_etudiants/Mm_morpion`. La structure de `position` est :

- `position.X` = vecteur contenant les positions des croix (l'ordinateur)
- `position.O` = vecteur contenant les positions des ronds (le joueur)
- `position.trait` = vrai si c'est à l'ordinateur de jouer
  La fonction `fournir_note` rend les valeurs: 2 si l'ordinateur gagne, 0 si l'utilisateur gagne, 1 dans les autres cas. Les positions sont données sous forme d'un seul entier correspondant au numéro de

Figure 6: Morpion.

Une fois les deux routines écrites et vérifiées, vous pouvez continuer le projet en suivant la piste qui vous inspire le plus : par exemple, introduire l'accélérateur $\alpha\beta$, placer un horizon dans la fonction `minmax`, écrire les fonctions `fournir_coups` et `fournir_note` correspondant à un autre problème, etc.

2. **IA, sujet libre : fonction heuristique** (en collaboration avec le groupe choisissant le sujet 1)

   L'objectif est d'écrire la fonction heuristique correspondant à un jeu (ou problème) à somme nulle et à deux joueurs (ou intervenants).

   Vous devrez donc créer les fonctions suivantes :

   - `liste_coups = fournir_coups(position)`, où `position` est un enregistrement décrivant la position du jeu et `liste_coups` est un vecteur d'enregistrements de même structure que `position` et que `coup`, qui contient tous les coups possibles.
   - `note = fournir_note(position)`, où `note` est la note d'une position.
   - `afficher_position(position)`, affichage à l'écran.

Vous devrez également créer le programme principal. Par contre, la routine effectuant le minmax vous sera donnée par le groupe 1.

Vous pouvez choisir de résoudre un problème non symétrique, au sens où les deux intervenants ne disposent pas des mêmes moyens d'action. Par exemple, un conducteur veut trouver le parcours le plus rapide entre 2 points d'une ville en considérant le pire des cas concernant le phasage des feux tricolores. Dans ce cas, l'un des intervenants est le conducteur et l'autre est la ville qui cherchera à maximiser le temps de parcours en jouant sur le phasage. Pour les problèmes non-symétrique, la fonction `liste_coups = fournir_coups(position,trait)` contient un paramètre supplémentaire: le trait indique l'intervenant dont c'est le tour de jouer (par exemple, trait=vrai pour le conducteur et faux pour la ville).

3. Réseau de neurones, sujet libre : toolbox Matlab

Sujet totalement libre. Il s'agit d'apprendre par vous-même à utiliser la toolbox *Neural Networks* de Matlab et de l'appliquer à un problème que vous aurez choisi.

4. Réseau de neurones : reconnaissance des fonctions alcools

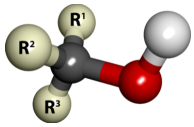

Figure 7: Fonction alcool.

Codez le réseau de neurones du TD *Réseaux de neurones linéaires* en utilisant, dans le programme, le même formalisme matriciel. Dans un premier temps, on considère que **b** est un vecteur binaire (constitué de 0 et de 1) et que les neurones comportent un comparateur de type Heaviside $H(y) = 1$ si $y > 0$, $H(y) = 0$ sinon. Le réseau sera optimisé pour discriminer les mots de 4 lettres contenant au moins une fois lettre "a" ($y = 1$) de ceux ne la contenant pas ($y = 0$). Pour créer des vecteurs **b** à partir de mots, on utilise la fonction Matlab suivante :

`b = reshape(de2bi(double(mot)),4*7,1)`

`double` permet de trouver les codes ascii des lettres de `mot`.

`char(i)` donne le character dont le code ascii est `i`.

Pour entraîner le réseau, on procède de la manière suivante:

- on initialise les matrices `We`, `Wc` et `Ws` à des valeurs quelconques
- on boucle sur des mots aléatoires contenant ou pas la lettre "a".
- pour chaque mot on construit son vecteur `b` et on calcule la sortie `y` du réseau
- si `y` n'a pas la valeur attendue :
  - si `y`=1 alors, pour tous les neurones j "allumés" (c'est-à-dire tels que $y_j = 1$) :
    - augmenter d'une unité leur biais θ_j
    - diminuer d'une unité tous les poids w_{ij} tels que $x_i = 1$
  - si `y`=0 alors, pour tous les neurones j "éteints" :
    - diminuer d'une unité leur biais θ_j
    - augmenter d'une unité tous les poids w_{ij} tels que $x_i = 1$
- on passe au mot suivant

Utiliser autant que possible le formalisme matriciel pour simplifier et accélérer le code. Etudiez le fonctionnement de ce réseau de neurones puis appliquez-le à la reconnaissance d'autres types d'entrées que vous choisirez, par exemple, la reconnaissance de la fonction alcool dans des formules condensées.

5. Algorithme génétique : dépollution des hydrocarbures

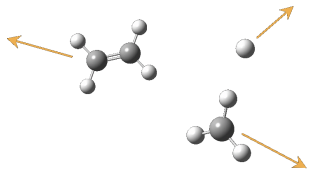

Figure 8: Fragmentation moléculaire.

Physico-chimie de la fragmentation On souhaite fragmenter des grosses molécules initiales d'hydrocarbure C_nH_m en molécules plus petites que le propane, en général moins polluantes (mais ce n'est pas le cas du propène, par exemple).

$$X_{\text{ini}} \longrightarrow \sum_{i=1}^{M} X_i$$

Pour cela, on excite ces hydrocarbures (par exemple à l'aide de lasers, de plasmas ou de faisceaux d'électrons). L'énergie E déposée dans la molécule se retrouve dans les termes suivants :

- $\Delta H = \sum_{i=1}^{M} H_{\text{th}}(i) - H_{\text{th}}(\text{ini})$ enthalpie de dissociation. On appelle multiplicité le nombre M de fragments, ini est le gros hydrocarbure initial et i le numero du fragment. C'est l'énergie dépensée pour fragmenter l'hydrocarbure initial (briser ses liaisons).
- E_{cin} la sommes des énergies cinétiques (translation et rotation) de tous les fragments.
- $E^* = \sum_{i=1}^{M} E^*(i)$ l'énergie d'excitation des fragments. Chacune des énergies $E^*(i)$ doit être plus petite que $\Delta H_{\text{min}}(i)$, sinon le fragment est trop excité et il se fragmente à son tour.

L'énergie cinétique et l'énergie d'excitation de la molécule initiale sont négligeables.

Si l'énergie E est trop petite (inférieure à $\Delta H_{\text{min}}(\text{ini})$), le gros hydrocarbure est simplement excité, mais il ne se fragmente pas. Pour E légèrement supérieure à $\Delta H_{\text{min}}(\text{ini})$, une liaison peut se casser, etc . A très hautes valeurs de E, l'hydrocarbure est atomisé (fragmenté en atomes indépendants). Entre les deux, l'hydrocarbure possède de nombreuses voies de fragmentation. Chacune des voies v est caractérisée par un poids W (dépendant de E) qui peut être calculé grâce à la physique statistique :

$$W(v,E) = W_{\text{elec}}(\mathbf{l},\mathbf{o})\ W_{\text{comb}}(\mathbf{n},\mathbf{m})\ W_{\text{ener}}(v,E)$$

- $W_{\text{elec}}(\mathbf{l},\mathbf{o}) = \prod_i (2\,l_i+1)\,(2\,o_i+1)$ est le poids électronique qui dépend des multiplicités de spin l_i et des multiplicités orbitélaires o_i.
- $W_{\text{comb}}(\mathbf{n},\mathbf{m}) = \frac{n(\text{ini})!\ m(\text{ini})!}{\prod_k k!^{M(k)}\ \prod_j N(j)!}$ est le poids combinatoires qui correspond au nombre de manières de répartir les carbones et hydrogènes initiaux dans les fragments
- $W_{\text{ener}}(v,E) = \frac{\Pi_\Delta\, E_{\text{cin}}^{\alpha-1}\,(2\,\pi)^\alpha}{\Gamma(\alpha)} \prod_{i=1}^{M} \frac{E_i^{*\,(f_{\nu i}-1)}}{\Gamma(f_{\nu j})\ \bar{\nu}_i}$ est le poids énergétique. Il représente toutes les manières de répartir l'énergie restant après fragmentation ($E_{\text{dispo}} = E - \Delta H = E_{\text{cin}} + E^*$), sur les différents degrés de liberté (excitation des fragments, énergie cinétique).

Les fonctions correspondantes vous sont données dans le folder `students_etudiants/AG_hydrocarbure`:

- `Welec(A)`

- `Wcomb(n_ini, m_ini, n, m)`
- `Wener(n_ini, m_ini, Edispo, A)`
- `Wcons(n_ini, m_ini, n, m)` vous permet de vérifier la conservation des nombres de carbones et d'hydrogènes. Elle vaut 1 si les conservations sont respectées, 2 s'il y a trop de C ou trop de H et 0 sinon.

La probabilité d'une voie de fragmentation est proportionnelle à son poids. Pour les énergies intermédiaires, le nombre de voies possibles est très grand mais les voies ayant une probabilité non négligeable sont très peu nombreuses (parfois une seule). Le projet consiste à trouver, grâce à un algorithme génétique, ces voies de fragmentation dominantes et leurs probabilités, en fonction de l'énergie.

Codage Un *chromosome* est ici une voie de fragmentation, les *gènes* sont les fragments (X_i) qui la composent. Un chromosome doit respecter les règles de conservation du nombre de carbones et du nombre d'hydrogènes. Lors de la phase de *reproduction*, le code doit tirer aléatoirement:

- soit un fragment du père,
- soit un fragment de la mère,
- soit un atome isolé,

jusqu'à ce que les règles de conservation soient respectées.

Si vous avez le temps, vous pouvez ajouter une phase de *mutation* que vous inventerez.

Pour charger les caractéristiques de tous les fragments possibles (il y en a 58), votre code doit débuter par la commande:

```
load('students_etudiants/AG_hydrocarbure/data.mat');
```

Vous disposerez ainsi des informations suivantes (les valeurs entre parenthèse indiquent la taille des tableaux):

- `n(58)`
- `m(58)`
- `formule(58)`
- `spin(58)` ('s' = singlet, 'd' = doublet, 't' = triplet)
- `geom(58)` ('l' = linéaire, 'c' = cycle de 3 carbones)
- `Hth(58)`
- `A(58,14)` matrice contenant les informations nécessaires pour le calcul de W_{elec} et W_{ener}. Les seules lignes de `A` à passer aux routine `AG_Welec` et `AG_Wener` sont celles correspondant aux fragments de la voie considérée. Les trois premières lignes de la matrice `A` contiennent les atomes isolés (C singlet, C triplet, H doublet).

Un chromosome est un vecteur d'indices. Par exemple, un chromosome de C_6H_6 est $[45, 5, 5, 1]$, c'est-à-dire CH_2CHCH (singlet, linéaire) + 2 CH (doublet) + C (singlet). Vous pouvez choisir la taille `n, m` de votre hydrocarbure initial. Son enthalpie peut être calculée par: `Hth_ini = n*HC+m*HH;`

6. IA : Algorithme MinMax appliqué au 2048

La règle du jeu se trouve sur Wikipedia; taper "2048 (jeu vidéo)".

Le but de ce projet est de développer une méthode qui permette à l'ordinateur d'obtenir le score le plus élevé possible au 2048 (un tel code a obtenu le second prix du Matlab Central Exchange). Pour vous aider, les routines suivantes se trouvent dans le folder `students_etudiants/Mm_2048`:

Figure 9: Une partie de 2048 en cours.

- `liste_coups = fournir_coups(position,trait)` où: `position.M` est une matrice 4×4 contenant la valeur des tuiles (0 en l'absence de tuile), `trait` est vrai si le coups suivant consiste à glisser les tuiles et faux si le coup suivant consiste à faire apparaître un "2" ou un "4" sur la grille et la sortie `liste_coups` est la liste de toutes les `position` possibles après ce coup.
- `position = glisse(position,fleche)` où: `fleche` est une lettre ('g','d','h','b') indiquant dans quelle direction les tuiles doivent être glissées et `position.M` est la liste des grilles possibles (entre 0 et 4 grilles).
- `afficher_position(position)` : représente l'état du jeu dans un fenêtre graphique, voir Fig. 9.

Une bonne manière de commencer est d'écrire le programme qui vous permet de jouer au 2048. Pour détecter la flèche pressée, vous pouvez utiliser la routine:

- `fleche = pressee()` (qui arrêtera le programme si "s" est pressée).

L'ordinateur devra ensuite vous remplacer en tant que joueur. Pour cela vous devez écrire deux routines: l'une pour donner une note à une grille, l'autre est la routine de minmax. Pour simplifier grandement la programmation, il faut que minmax soit une fonction récursive, c'est-à-dire une fonction qui peut s'appeler elle-même (voir Wikipedia et la doc. de Matlab). Dans ce cas, le code ne devrait comporter qu'une vingtaine de lignes.

7. Algorithme génétique : sujet libre

Appliquez un algorithme génétique à un problème que vous aurez choisi.