TI301 – Algorithmique et Structures de données 2

Projet Informatique et Mathématiques

Ce projet est rédigé en commun avec le département de mathématiques

Etude de Graphes de Markov – PARTIE 3

Il est maintenant temps de nous intéresser aux propriétés de ces graphes pour les probabilités.

Notion de distribution

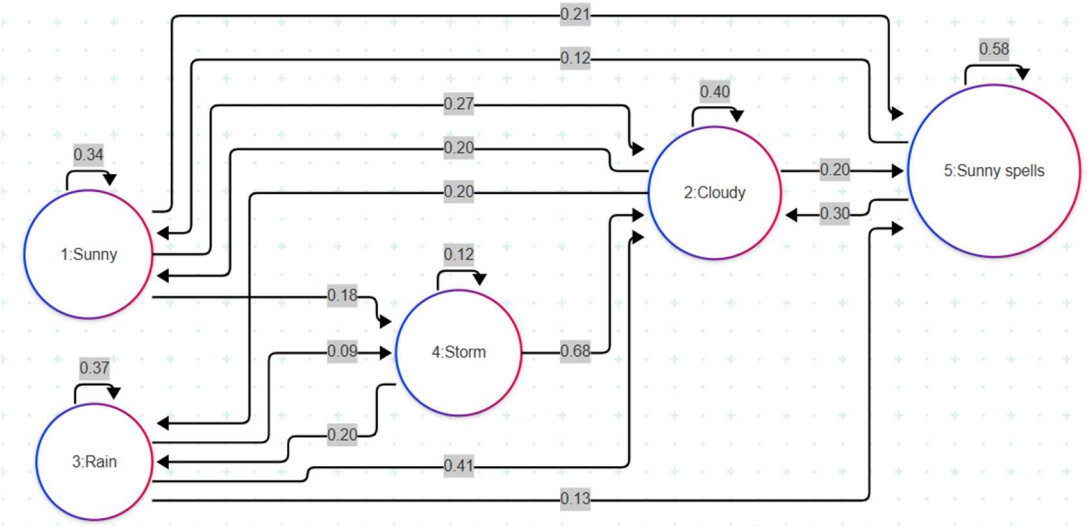
Pour le moment, nous avons associé des probabilités aux arêtes, indiquant quelle est la probabilité de passer d'un état à un autre.

Un système, à un instant donné, devrait se trouver dans un état défini en fonction d'un état de 'départ'. Cependant, comme les transitions sont probabilistes, il n'est pas possible d'affirmer que le système se trouvera dans 'tel état' après 'un certain nombre' de transitions.

En revanche, on peut indiquer, pour le système, quelles sont les probabilités qu'il se trouve dans un état donné à un instant t, en fonction d'un 'état' de départ.

Illustrations et exemples avec une graphe de Markov pour la météo

Soit le graphe de Markov suivant, illustrant (de manière totalement arbitraire), les évolutions de la météo jour après jour : chaque état représente la météo d'une journée, chaque transition indique la probabilité de passer d'un état à l'autre. Le 'temps discret' utilisé est une journée : on 'calcule' les évolutions d'un jour J à un jour $J+1$, puis de $J+1$ à $J+2$, et ainsi de suite.



Une distribution représente, sur ce graphe, les probabilités d'être dans un état donné à un instant donné. Une distribution est la répartition de probabilités sur tous les états : sa somme doit être égale à 1. Il s'agit d'un vecteur ligne, où chaque coordonnée indique la probabilité de chaque état du graphe.

La notation consacrée pour une distribution est : Π

Exemple

Aujourd'hui il faut beau (« Sunny ») : indique que la probabilité de se trouver dans l'état 1 est égale à 1, toutes les autres valant 0.

La distribution associée est alors $\Pi = (1 \quad 0 \quad 0 \quad 0 \quad 0)$

Voici un petit tableau pour vous aider à interpréter cette distribution

Numéro de l'état	1	2	3	4	5
Météo associée	Sunny	Cloudy	Rain	Storm	Sunny spells
Probabilité	1	0	0	0	0

Calcul de l'évolution en temps discret d'une distribution

En étudiant les distributions, les graphes de Markov nous permettent de répondre aux questions du type :

1. « Quelle est la probabilité que le temps soit nuageux (**Cloudy**) dans 3 jours s'il fait beau (**Sunny**) aujourd'hui » ?
2. « Quelles sont les probabilités que la météo soit dans tel état dans une semaine (7 jours) sachant qu'il pleut (**Rain**) aujourd'hui ? »
3. « Atteint-on des probabilités indépendantes de la distribution de départ au bout d'un certain temps ? » (en termes mathématiques, existe-t-il une distribution stationnaire ?)

Principes de calcul

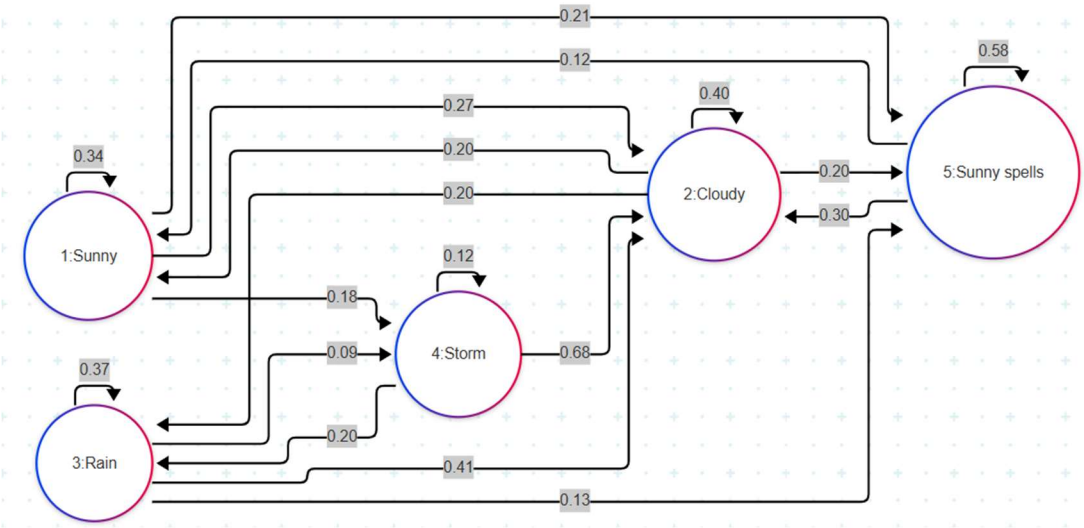
Le graphe de Markov étant représenté par une matrice M, et connaissant une distribution initiale Π_0, on calcule l'évolution de la distribution $\Pi_1 = \Pi_0 . M$

On peut ensuite recommencer, et calculer : $\Pi_2 = \Pi_1 . M = (\Pi_0 . M) . M = \Pi_0 . M^2$

De proche en proche, on peut donc calculer l'évolution de la distribution initiale Π_0 après *n* étapes : $\Pi_n = \Pi_0 . M^n$

Illustration avec le graphe 'Météo'

La matrice M indique les probabilités de passage d'un état à un autre – voici la représentation matricielle du graphe pris en exemple.



$$M = \begin{pmatrix} 0.34 & 0.27 & 0 & 0.18 & 0.21 \\ 0.20 & 0.40 & 0.20 & 0 & 0.20 \\ 0 & 0.41 & 0.37 & 0.09 & 0.13 \\ 0 & 0.68 & 0.20 & 0.12 & 0 \\ 0.12 & 0.30 & 0 & 0 & 0.58 \end{pmatrix}$$

Quels calculs devons-nous effectuer pour répondre aux questions posées ?

1. « Quelle est la probabilité que le temps soit nuageux (**Cloudy**) dans 3 jours s'il fait beau (**Sunny**) aujourd'hui » ?
2. « Quelles sont les probabilités que la météo soit dans tel état dans une semaine (7 jours) sachant qu'il pleut (**Rain**) aujourd'hui ? »
3. « Atteint-on des probabilités indépendantes de la distribution de départ au bout d'un certain temps ? » (en termes mathématiques, existe-t-il une distribution stationnaire ?)

<u>Pour la question 1</u> : la distribution initiale est $\Pi_0 = (1 \quad 0 \quad 0 \quad 0 \quad 0)$, on cherche à calculer :

$\Pi_3 = \Pi_0 . M^3$, (3 pour 'dans 3 jours') et on obtient :

$$M^3 = \begin{pmatrix} 0.17 & 0.37 & 0.13 & 0.05 & 0.27 \\ 0.16 & 0.37 & 0.14 & 0.05 & 0.28 \\ 0.14 & 0.38 & 0.18 & 0.04 & 0.25 \\ 0.15 & 0.38 & 0.18 & 0.05 & 0.24 \\ 0.17 & 0.34 & 0.09 & 0.04 & 0.35 \end{pmatrix}$$

Puisque $\Pi_0 = (1 \quad 0 \quad 0 \quad 0 \quad 0)$, $\Pi_3 = (0.17 \quad 0.37 \quad 0.13 \quad 0.05 \quad 0.27)$

Numéro de l'état	1	2	3	4	5
Météo associée	Sunny	Cloudy	Rain	Storm	Sunny spells
Probabilité au bout de 3 jours	0.17	0.37	0.13	0.05	0.27

Ainsi, s'il fait beau (**Sunny**) aujourd'hui, la probabilité que le temps soit nuageux (**Cloudy**) dans 3 jours est 0.37 = 37 %.

Pour la question 2 : la distribution initiale est $\Pi_0 = (0 \quad 0 \quad 1 \quad 0 \quad 0)$, (il pleut, l'état est '**Rain**') on cherche à calculer : $\Pi_7 = \Pi_0 . M^7$, (7 pour calculer 'dans 7 jours') et on obtient :

$$M^7 = \begin{pmatrix} 0.16 & 0.36 & 0.13 & 0.05 & 0.29 \\ 0.16 & 0.36 & 0.13 & 0.05 & 0.29 \\ 0.16 & 0.36 & 0.13 & 0.05 & 0.29 \\ 0.16 & 0.36 & 0.13 & 0.05 & 0.29 \\ 0.16 & 0.36 & 0.13 & 0.05 & 0.30 \end{pmatrix}$$

Puisque $\Pi_0 = (0 \quad 0 \quad 1 \quad 0 \quad 0)$, $\Pi_7 = (0.16 \quad 0.36 \quad 0.13 \quad 0.05 \quad 0.29)$

Ainsi, s'il pleut aujourd'hui : **dans une semaine**, il y 16% de chances qu'il fasse beau, 36% de chances que le temps soit nuageux, 13% de chances qu'il pleuve, 5% de chances que le temps soit orageux, et 29% de chances qu'il y ait des éclaircies.

Pour la question 3 : on constate déjà que, dans la matrice M^7, toutes les lignes sont égales (modulo les arrondis), et ainsi, quel que soit l'état de départ (=le temps qu'il fait aujourd'hui), on arrive au même résultat :

Quelle que soit la météo aujourd'hui, dans une semaine : il y 16% de chances qu'il fasse beau, 36% de chances que le temps soit nuageux, 13% de chances qu'il pleuve, 5% de chances que le temps soit orageux, et 29% de chances qu'il y ait des éclaircies. Cela est aussi dû au fait que c'est une représentation très simplifiée de l'évolution de la météo.

La distribution, nommée $\Pi^* = (0.16 \quad 0.36 \quad 0.13 \quad 0.05 \quad 0.29)$ est dite stationnaire. Il n'y aura plus d'évolution des probabilités, et on a donc : $\Pi^* . M = \Pi^*$

A vous de jouer

Vous pouvez utiliser des outils d'IA pour cette partie, mais devrez expliquer :

- ✓ **Si vous utilisé de tels outils ;**
- ✓ **Le prompt que vous avez utilisé ;**
- ✓ **Le code obtenu.**

Ces questions vous seront posées lors de la soutenance

Étape 1 : calculs matriciels

En utilisant des fichiers **`matrix.c`** / **`matrix.h`** , définissez les fonctions suivantes :

- ✓ Une fonction qui, à partir d'une liste d'adjacence pour un graphe à n états, crée une matrice $n\ x\ n$ remplie avec les probabilités de passage entre états ;
- ✓ Une fonction qui crée une matrice $n\ x\ n$ remplie avec la valeur 0 ;
- ✓ Une fonction qui recopie les valeurs d'une matrice dans une autre de même taille ;
- ✓ Une fonction de multiplication de deux matrices $n\ x\ n$;
- ✓ Une fonction qui calcule 'la différence' entre deux matrices M et N : $diff(M,N) = \sum_i \sum_j |m_{ij} - n_{ij}|$ – somme des valeurs absolues des différences entre les coefficients des matrices.

Validation de l'étape 1

- ✓ Affichage de la matrice M associée à l'exemple météo (fichier **`exemple_meteo.txt`**) ;
- ✓ Calcul de M^3, vous devez obtenir le même résultat que celui présenté en exemple ;
- ✓ Calcul de M^7, vous devez obtenir le même résultat que celui présenté en exemple ;
- ✓ Avec les fichiers exemple : calculer M^n, pour laquelle la différence entre M^n et M^{n-1} est inférieure à $\varepsilon = 0.01$. (attention, sur certains exemples, ce critère peut ne pas fonctionner : indiquez quels sont ces exemples).

Étape 2 : propriétés des graphes de Markov

Les graphes de Markov présentent les propriétés suivantes (sans démonstration)

- ✓ Un graphe irréductible (une seule classe) possède une unique distribution limite

- ✓ Pour un graphe non irréductible (plusieurs classes) :
  - Les classes **transitoires** ont une distribution limite nulle (les probabilités sont toutes égales à 0) ;
  - Les classes **persistantes** ont chacune une distribution limite ;

Dans les fichiers **`matrix.c`** / **`matrix.h`**, ajoutez la fonction suivante (en utilisant les éléments de la partie 2) :

```
/**
 * @brief Extracts a submatrix corresponding to a specific
component of a graph partition.
 *
 * @param matrix The original adjacency matrix of the graph.
 * @param part The partition of the graph into strongly
connected components.
 * @param compo_index The index of the component to extract.
 * @return t_matrix The submatrix corresponding to the
specified component.
 */
t_matrix subMatrix(t_matrix matrix, t_partition part, int
compo_index);
```

Dans cette 'sous-matrice', on ne garde que les lignes et colonnes des sommets appartenant à une composante donnée.

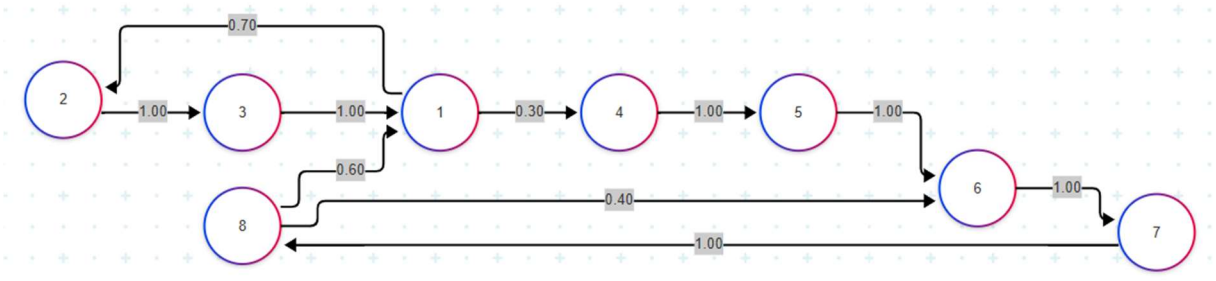
Cela vous permettra de créer une 'sous-matrice' pour une classe donnée : en calculant les puissances de ces sous-matrices, vous obtiendrez les distributions stationnaires par classes.

Validation de l'étape 2

Vous obtenez les distributions **stationnaires pour chacune des classes d'un graphe** (test sur les fichiers exemples fournis).

Étape 3 (défi en bonus)

Périodicité des classes : certaines classes n'ont pas de distribution limite, mais **plusieurs** distributions stationnaires (et périodiques), car les probabilités évoluent de manière 'cyclique'. Voici un exemple d'un graphe périodique :



Ce graphe est irréductible (une seule classe – vérifiez-le avec votre programme), mais possède une 'période' . En partant de tout sommet, on peut y revenir au bout de 3 étapes (mais ni une étape, ni 2), ou au bout de 6 étapes.

Voici donc le défi à relever - je vous fournis le code brut de calcul de période pour une classe (code à adapter en fonction de vos structures de données et des fonctions que vous avez écrites).

```
int gcd(int *vals, int nbvals) {
    if (nbvals == 0) return 0;
    int result = vals[0];
    for (int i = 1; i < nbvals; i++) {
        int a = result;
        int b = vals[i];
        while (b != 0) {
            int temp = b;
            b = a % b;
            a = temp;
        }
        result = a;
    }
    return result;
}

int getPeriod(t_matrix sub_matrix)
{
    int n = sub_matrix.rows;
    int *periods = (int *)malloc(n * sizeof(int));
    int period_count = 0;
    int cpt = 1;
    t_matrix power_matrix = createEmptyMatrix(n);
    t_matrix result_matrix = createEmptyMatrix(n);
    copyMatrix(power_matrix, sub_matrix);

    for (cpt = 1; cpt <= n; cpt++)
    {
        int diag_nonzero = 0;
        for (int i = 0; i < n; i++)
        {
            if (power_matrix.data[i][i] > 0.0f)
            {
                diag_nonzero = 1;
            }
        }
        if (diag_nonzero) {
            periods[period_count] = cpt;
            period_count++;
        }
        multiplyMatrices(power_matrix, sub_matrix, result_matrix);
        copyMatrix(power_matrix, result_matrix);
    }

    return gcd(periods, period_count);
}
```

Défi numéro 1

Commentez et expliquez ce code, intégrez-le à votre programme.

Défi numéro 2

calculez les périodes des classes du graphe (la période sera alors la même pour tous les sommets appartenant à cette classe), puis trouver les distributions stationnaires associées.