INFO-F-308 Projets d'informatique 3 transdisciplinaire

–

?? ?? 2017

Contents

Chapter 1

Introduction

Une machine est déterministe. Elle ne peut donc pas générer de nombres aléatoires. Cependant, lorsque ceci lui est demandé, elle peut simuler un nombre aléatoire, appelé nombre pseudo-aléatoire. Il existe différentes techniques de génération de nombres pseudo-aléatoires dont la plupart sont basées sur l'observation de phénomènes imprévisibles pour l'Homme, et qui sont donc considérés comme « aléatoires », comme par exemple: le cours de la bourse, la psychanalyse, une lampe à lave, un double pendule...

Entropie

Une de ces méthodes se base sur l'accumulation d'entropie. Le mot « entropie » caractérise le degré d'imprédictibilité ou de « désordre » de l'information contenue dans un système. Ce terme a été introduit par Rudolf Clausius, physicien allemand, en 1865. L'entropie est utilisée dans de nombreux domaines de la science, telle la physique (dans la thermodynamique) et les mathématiques.

La méthode d'accumulation d'entropie consiste à rassembler un certain nombre d'événements aléatoires. Grâce à une fonction de transformation, cet ensemble d'événements aléatoires, à un moment précis, générera un nombre pseudo-aléatoire.

Il existe également d'autres méthodes, telles que **les méthodes physiques**. En effet, certains évènements physiques imprévisibles et aléatoires permettent d'en générer directement, toujours grâce à une fonction de transformation, un nombre aléatoire. Il faut cependant faire attention à ne pas modifier le système physique pour que ses résultats ne soient biaisés lors de la mesure.

La méthode que nous avons choisi d'implémenter pour générer des nombres aléatoires fait partie de la deuxième catégorie: c'est une méthode physique. Elle se base sur une lampe à lave. Une lampe à lave est un objet décoratif contenant un liquide transparent dans lequel évolue de manière imprévisible une cire fondue. Cette imprévisibilité lui offre un caractère aléatoire, de par le comportement cahotique des gouttes de cire. Celui-ci est du à différents

phénomènes physiques comme la température, densité, les turbulances etc.

Chaos

Un système dynamique est cahotique quand la prédiction du comportement futur devient impossible au delà d'un certain intervalle temporel, malgré la connaissance des méchanismes de fonctionnement et indépendamment de la précision de la condition initiale. Plus connu sous l'appellation d'"effet papillon", ce comportement est du à l'instabilité non linéaire du système, qui mène à l'amplification très rapide des erreurs des mesures initiales.

Notion d'algorithme

Définition
Selon le dictionnaire "Larousse", un algorithme est un "ensemble de règles opératoires dont l'application permet de résoudre un problème énoncé au moyen d'un nombre fini d'opérations. Un algorithme peut être traduit, grâce à un langage de programmation, en un programme exécutable par un ordinateur". Lorsqu'elle génère un nombre aléatoires, une machine utilise donc un algorithme prédéfini qui lui dicte les étapes à suivre afin d'obtenir ce nombre.

Sysème logique

Un système logique est une branche des **systèmes formels**.

Un système formel est un quadruplet S = (A, γ, AX, B) où :

1. A est un alphabet, ensemble fini de symboles
2. γ est un procédé de construction de mots
3. AX est un ensemble fini ou dénombrable d'axiomes (mots particuliers)
4. R est un ensemble fini de règles de déduction

Les systèmes logiques concernent les déductions logiques. Une déduction logique relie des propositions appelées "prémisses" à une conclusion, en conservant la vérité.

Proposition

Selon le dictionnaire Larousse, une proposition "en logique est une assertion fondamentale jugée trop élémentaire pour être qualifiée de théorème. (Sa caractéristique essentielle est d'être susceptible de recevoir une valeur de vérité.)". Une proposition est dite valide si elle est vraie pour toute interprétation de ses variables.
??????????????????????????????????

Chapter 2

Machine de Turing

Une machine de Turing est un concept abstrait (objet mathématique) surtout utilisé en informatique théorique. Elle fut inventée en 1936 par Alan Turing (mathématicien et cryptologue britannique) en vue de donner une définition au concept d'algorithme, ainsi que pour répondre au problème de décidabilité: « Existe-t-il un algorithme capable de décider de la validité d'une proposition énoncée dans un système logique? ». Aujourd'hui, elle est encore fort utilisée dans le domaine de la calculabilité.

Thèse Church-Turing

La machine de Turing est un modèle universel qui peut calculer ce que n'importe quelle machine peut calculer. Dès lors, un problème peut être résolu par un algorithme si et seulement si il peut être résolu par une machine de Turing. De ce fait, ce qui n'est pas calculable par une machine de Turing, ne sera pas calculable par n'importe quel algorithme.
Ceci est une thèse et non pas un théorème car il est impossible de démontrer (!) l'idée défendue. Un contre-exemple pourrait montrer que la thèse est fausse, cependant un tel exemple n'a jamais été trouvé.

2.1 Composantes

La machine de Turing est la machine la plus élémentaire destinée à mettre en oeuvre des mécanismes de calcul. C'est un automate fini capable de lire et d'écrire des données sur un ruban.

Selon Sipser : $http://www.cis.upenn.edu/\ matuszek/cit596-2012/NewPages/turing-machine-definitions.html$

La machine de Turing possède plusieurs composantes principales:

1. Un ruban infini: celui-ci est divisé en cases ou cellules consécutives, chacune contenant un symbole d'un alphabet défini et fini appelé « alphabet de travail ». On suppose que le ruban est de longueur infinie, vers la gauche comme vers la droite. Cela signifie qu'on considère que la machine a toujours assez de longueur de ruban pour son exécution. On considère

également que les cases non encore écrites contiennent le symbole blanc qui est un symbole spécial.

2. Une tête de lecture/écriture: elle permet de lire et écrire les symboles sur le ruban ainsi que de se déplacer sur le ruban (vers la gauche ou vers la droite).

3. Un ensemble fini d'états ainsi qu'un registre d'état qui mémorise l'état courant de la machine de Turing. Il existe un état spécial appelé « état de départ » qui correspond à l'état initial de la machine, avant son exécution, ainsi qu'un état spécial appelé « état terminal ».

4. Une table de transition qui contient toutes les fonctions de transition: celle-ci indique à la machine quel symbole écrire sur le ruban, comment déplacer la tête de lecture et quel est le nouvel état. Si aucune action n'existe, la machine s'arrête. En d'autres mots, la table de transition indique pour chaque couple (état_interne, caractère_courant) les nouvelles valeurs pour celui-ci, ainsi que le déplacement de la tête de lecture/écriture. Dans cette table, chaque couple est associé à un triplet (nouvel_état_interne, nouveau_caractère, déplacement).

2.2 Fonctionnement

Le fonctionnement de la machine de Turing est discret et logique. Les actions prises par la machine à chaque étape sont paramétrées par le symbole lu et l'état de la machine. Les étapes sont:

1. Initialisation du ruban: le ruban est initialisé avec la séquence de caractères correspondant aux données d'entrée. La tête de lecture/écriture est positionnée sur la première case du ruban et l'état interne est positionné à sa valeur initiale.

2. Avancement: le triplet (nouvel_état_interne, nouveau_caractère, déplacement) est utilisé pour mettre à jour l'état interne et le caractère courant avant d'effectuer le déplacement de la tête de lecture/écriture, tant que le couple courant (état_interne, caractère_courant) se trouve dans la table de transition.

3. Arrêt de la machine: si le couple (état_interne, caractère_courant) ne se trouve pas dans la table de transition, la machine s'arrête. La séquence de caractères stockée à ce moment précis sur le ruban est considérée comme le résultat du traitement.

La représentation canonique est une représentation uniforme des machines de Turing qui consiste à imposer la codification binaire des entrées et sorties et l'absence de caractères dans une case par le caractère ϵ.

Chapter 3

Complexité de Kolmogorov

L'idée sur laquelle Kolmogorov fonde sa théorie est simple : « un objet est complexe quand il n'en existe pas de description courte ».
La complexité de Kolmogorov permet de mesurer la quantité d'information contenue dans un mot. Celle-ci est définie comme la taille du plus petit algorithme qui engendre ce mot. Cette complexité a été découverte par le mathématicien russe Andreï Kolmogorov en 1963. Avant de définir avec précision ce concept, nous présenterons un exemple intuitif de la complexité de Kolmogorov.
Exemple:
Chaine 1: abababababababab
Chaine 2: oiuytrdcvbnhgfredfmma
Il est intuitif de dire que la première chaine peut être résumée comme neuf fois la chaine « ab », alors que la deuxième chaine ne possède pas de version compressée.

Définition formelle
Soit T, une machine de Turing et B, son alphabet 0,1. L'ensemble des mots finis sur cet alphabet est noté B*. Soit une chaine donnée x.
Soit f:B*->B*, une fonction calculable. On définit la complexité de x $\in$ B* par rapport à f comme étant la valeur
K_f(x)= { min(t) tel que f(t)=x ou inf si un tel t n'existe pas }.

Dès lors, t est une description de x par rapport à f. Cette définition possède cependant un inconvénient: la complexité dépend du choix de la fonction f. Tout de même, nous savons qu'il existe une fonction que nous noterons f_0 appelée fonction optimale, telle que quelle que soit une autre fonction f, il existe une constante C telle que:
$\forall$ x $K_{f0}(x) \leq K_f(x) + C$
Aucun procédé ne calcule exactement K(s). K(s) n'est pas calculable. Nous savons que cette complexité existe, mais nous ne savons la calculer exactement.

Comme vu dans l'exemple précédent, la complexité de Kolmogorov est étroitement liée à la notion de compression de données. La version comprimée d'un texte peut se définir comme une sorte de programme qui engendre ce même texte. Ainsi, la complexité de Kolmogorov d'une chaine représente la taille

de sa meilleure version comprimée. Nous pouvons donc utiliser la recherche d'une méthode de compression sans perte comme un outil qui nous permettrait d'approximer la complexité de Kolmogorov.