2I013 – Projet

Sujets de projets

Sujet 1 : Recommandation et analyse de sous-titres

- Encadrants : Nicolas Baskiotis, Vincent Guigue (`prenom.nom@lip6.fr`)
- Titre : Recommandation et analyse de sous-titres
- Nombre d'étudiants : 2 ou 3
- Description : Les algorithmes de recommandation sont au coeur de nombreux produits industriels : recommandation d'amis dans les réseaux sociaux, de produits dans les sites marchands, de vidéos dans les sites de partage ou de VOD ... Leur objectif est de dépasser les algorithmes usuels de recherche d'informations en proposant des suggestions directement à partir du profil d'un individu, sans forcément attendre une requête explicite sur le contenu. Les algorithmes de recommandation se divisent en deux familles : les algorithmes basés sur le contenu, qui recommandent des choses proches des items déjà visités et les algorithmes de filtrage collaboratif, basés sur la recommandation de produits appréciés par les personnes qui aiment les mêmes choses que la personne cible.
  Nous proposons d'explorer ces différentes stratégies sur la recommandation de séries TV, en exploitant une base de sous-titres en guise de contenu et un ensemble d'avis d'utilisateurs. Le projet se déroulera selon le plan suivant :
  - Prise en main des données textuelles et des outils de traitement de la langue
  - Prise en main des algorithmes de filtrage collaboratif
  - Comparaison des stratégies de recommandation
  - Scrapping de données pour l'enrichissement du contenu et des interactions, et l'évaluation des méthodes mises en oeuvre.

  Il est recommandé d'avoir une connaissance du langage python.

Sujet 2 : Vectorisation, arithmétique modulaire et parallélisme

- Encadrant(e) : Mohab SAFEY EL DIN (`mohab.safey@lip6.fr`)
- Titre : Vectorisation, arithmétique modulaire et parallélisme
- Nombre d'étudiants : 2
- Description : Les algorithmes de la cryptographie, la géométrie algorithmique ou encore du calcul formel reposent en grande partie sur des calculs dits modulaires, c'est-à-dire effectués modulo des nombres premiers de taille binaire inférieure au mot machine. Cela permet d'utiliser les couches arithmétiques au niveau CPU sans surcouche logicielle (qui est 10 à 100 fois plus lente).
  Un enjeu essentiel pour de nombreuses applications est alors d'exploiter au maximum les capacités des serveurs de calculs. Deux options (non exclusives) sont alors possibles. On peut, dans un premier temps, utiliser des instructions de vectorisation qui permettent d'effectuer «plusieurs unités de calculs en même temps». Dans ce projet, on s'intéressera à cette approche ; on tentera notamment d'implanter un algorithme d'Euclide sur des polynômes à coefficients

modulaires en exploitant les instructions AVX2 (voire AVX512 si on parvient à donner un accès aux serveurs de calcul pour les étudiants).
Une seconde option est l'utilisation du parallélisme, via le multi-threading. Si le temps le permet (et seulement dans ce cas), on s'intéressera à des implantations d'algorithmes de l'algèbre linéaire qui, en plus d'exploiter les instructions de vectorisation, intégreront du multi-threading.

Sujet 3 : résolution d'un systèmes linéaires structurés

- Encadrant : Jérémy Berthomieu (`jeremy.berthomieu@lip6.fr`)
- Titre : résolution d'un systèmes linéaires structurés
- Nombre d'étudiants : 2 ou 3
- Description : En calcul formel, nous nous intéressons à l'étude et aux développement d'algorithmes et de logiciels pour manipuler des objets de nature mathématique. En particulier, l'un des objectifs est de modéliser différents problèmes à l'aide de "briques de base" comme la multiplication d'entiers, de polynômes, de matrices et de trouver des algorithmes efficaces pour ces briques de base.
  Un problème classique est la résolution de systèmes linéaires structurés. Résoudre de tels systèmes linéaires trouvent de nombreuses applications comme la résolution d'un système linéaire creux, la résolution d'un système polynomial à plusieurs variables ou la correction d'erreurs en théorie des codes...
  Dans une première partie du projet, on montrera comment l'algorithme de Wiedemann ramène la résolution d'un système linéaire creux à la résolution d'un système linéaire structuré. Dans une seconde partie du projet, on montrera comment la résolution d'un tel système linéaire se ramène à un problème de calcul de relations de récurrence d'une suite et son lien avec l'algorithme d'Euclide étendu sur les polynômes. Enfin, on verra comment relier la complexité de l'algorithme d'Euclide étendu sur les polynômes à celle de la multiplication de polynômes. En particulier, comment bénéficier des algorithmes rapides de multiplications dans l'algorithme d'Euclide.

Sujet 4 : Implantation d'un algorithme de complexité paramétrée pour un problème d'ordonnancement classique

- Encadrant(e) : Alix Munier (`Alix.Munier@lip6.fr`)
- Titre : Implantation d'un algorithme de complexité paramétrée pour un problème d'ordonnancement classique.
- Nombre d'étudiants : 2 ou 3
- Description : On considère un problème d'ordonnancement défini par un ensemble de n tâches de durée unitaire numérotées de 1 à n. Chaque tâche $i \in \{1, \ldots, n\}$ possède une date de disponibilité r_i et une date de fin d_i. Des contraintes de précédence entre les tâches sont exprimées sous la forme d'un graphe sans circuit G. Enfin, les tâches sont à réaliser sur m machines identiques. Le problème consiste à fixer les dates d'éxécutions des tâches de sorte à respecter l'ensemble des contraintes si cela est possible.
  Ce problème d'ordonnancement est difficile à résoudre. Le but de ce projet est d'impémenter et de tester un algorithme paramétré qui permet de calculer, si il existe, un ordonnancement réalisable pour ce problème.

L'algorithme à programmer consiste à énumérer et coder de manière astucieuse l'ensemble des solutions réealisables du problème. Il sera implémenté en Python et la bibliothèque NetworkX pour les graphes. Le paramètre correspond ici au nombre maximum de tâches qui peuvent être réalisées en même temps si on ne considère que les intervalles $[r_i, d_i]$. Pour une valeur fixée de ce paramètre, la complexité de l'ordonnancement est une fonction polynomiale du nombre de tâches.
La complexité de l'algorithme sera alors évaluée expérimentalement sur des instances aléatoires et comparée à la complexité théorique. Des expérimenations pourront également être effectuées pour des structures de graphe particulières.

Sujet 5 : Détecteur de Harris-Laplace

— Encadrant : Dominique Béréziat (`dominique.bereziat@lip6.fr`)
— Titre : Détecteur de Harris-Laplace
— Nombre d'étudiants : 2 ou 3
— Description : Le détecteur de Harris est un filtre qui localise les coins dans les images. Ce filtre est invariant par rotation mais pas par changement d'échelle ce qui le rend peu robuste. Le détecteur de Harris-Laplace est une version multi-échelles, et lui permet donc, en principe d'être invariant par changement d'échelle. Le travail demandé est : la lecture des articles décrivant les deux détecteurs, leur implantation (langage de programmation à la discrétion des étudiants), et leur expérimentation sur diverses images en évaluant notamment la performance, et la robustesse aux changements d'échelles.
Références bibliographiques :
— Harris, C. and Stephen, M. : A combined corner and edge detector, Alveyn Vision Conference 1988, `https://tinyurl.com/qpmcp7p`
— Mikolajczyk, K. and Schimid, C. : Scale & Affine Invariant Interest Point Detectors, in IJCV 2004, `https://tinyurl.com/yjcb6o8`.

Sujet 6 : Classifieurs à base de modèles graphiques probabilistes dans scikit-learn

— Encadrant(e) : Pierre-Henri Wuillemin (`pierre-henri.wuillemin@lip6.fr`)
— Titre : Classifieurs à base de modèles graphiques probabilistes dans scikit-learn
— Nombre d'étudiants : 2
— Description :
Les modèles graphiques probabilistes sont des outils de manipulation de distributions de probabilité jointes de grande taille. Ils permettent des calculs d'inférence et des apprentissages dans de hautes dimensions difficilement atteignables par d'autres outils probabilistes. Une utilisation particulière de ces modèles peut être de fournir un classifieur probabiliste complexe.
Étant donnée la technicité nécessaire à l'implémentation efficace de ces modèles, il n'existe pas (à notre connaissance) d'accès aisé à de tels classifieurs dans des librairies classiques de classification comme scikit-learn. Par ailleurs, pyAgrum, une librairie de manipulation de tels modèles graphiques probabilistes (réseaux bayésiens) est développée au LIP6 (https ://www.agrum.org) mais sans mettre particulièrement l'accent sur la classification.
Le but de ce stage est de proposer une interface entre la libraire pyAgrum et la librairie scikit-learn afin d'ajouter les modèles graphiques au spectre de classifieurs que sait traiter scikit-learn. Outre l'aspect technique de cet interfaçage, il s'agira également de développer

des algorithmes particuliers pour la création de modèles graphiques dédiés à la classification comme les NaiveBayes, les TAN et les couvertures de Markov.

Sujet 7 : Comparaison entre algorithmique combinatoire et programmation linéaire

- Encadrant(e) : Pierre Fouilhoux (`pierre.fouilhoux@lip6.fr`)
- Titre : Comparaison entre algorithmique combinatoire et programmation linéaire
- Nombre d'étudiants : 2
- Description : Lorsqu'on connaît un algorithme polynomial pour un problème combinatoire, on choisit fréquemment une bonne implémentation de cet algorithme. Il s'agit en général d'un algorithme combinatoire, c'est-à-dire une suite d'opérations simples sur les objets du problème. C'est le cas par exemple du fameux algorithme hongrois pour la recherche d'un couplage entre deux ensembles d'éléments. Il existe pourtant un autre type d'algorithmes basés sur la programmation linéaire et qui sont également polynomiaux, mais dont on ne connaît pas précisément la valeur de l'exposant du polynôme : les algorithmes basés sur la programmation linéaire (introduits dans le module L2 nommé MOCA mais non proposé aux étudiants du parcours). L'objectif de ce projet est de comparer expérimentalement les deux familles d'algorithmes sur des exemples concrets (couplage, couverture de sommets) afin de déterminer leurs efficacités.

Sujet 8 : Prédiction de liens dans les graphes

- Encadrant : Lionel Tabourier (`lionel.tabourier@lip6.fr`)
- Titre : prediction de liens dans les graphes
- Nombre d'étudiants : 2 ou 3
- Description :
  Une grande variété de données relationnelles sont représentées par des graphes, par exemple un réseau social dont les nœuds seraient des comptes et les liens des relations d'amitié sur le réseau. La structure de ces réseaux évolue dans le temps, et il est donc très pertinent de chercher à savoir quels liens sont suceptibles d'apparaître.
  On peut en partie déduire ces futures interactions de l'état actuel du réseau en décrivant sa structure. Le projet consiste à réaliser cette tâche de prédiction de liens en mesurant différents indicateurs de la structure du graphe pour comparer leurs performances sur divers jeux de données, Cela suppose de définir un protocole pour évaluer les performances sur une telle tâche. Pour ce faire, nous utiliserons le vocabulaire de la classification à deux classes sur les paires de nœuds du graphe, les classes étant "paire connectée" et "non-connectée". Les stagiaires examinerons également à quel point de tels indicateurs passent à l'échelle. Enfin, en fonction de l'avancement du travail, on pourra étudier comment il est possible d'améliorer les prédictions avec des techniques d'apprentissage non-supervisé ou supervisé.

Sujet 9 : Notions de centralité dans les graphes

- Encadrant : Lionel Tabourier (`lionel.tabourier@lip6.fr`)
- Titre : notions de centralité dans les graphes
- Nombre d'étudiants : 2 ou 3

— Description :
Les notions de centralité des nœuds d'un graphe cherchent à traduire l'idée d'importance que ce nœud a dans le fonctionnement d'un réseau, représenté par un graphe. Comme la notion d'importance peut elle-même avoir des significations très variées selon le système étudié, plusieurs définitions ont été proposées pour quantifier la centralité.
Par exemple, la centralité d'intermédiarité (*betweenness centrality*) quantifie la tendance d'un nœud à être localisé sur un plus court chemin entre deux autres nœuds du graphe. Ainsi, dans un réseau social, elle est souvent interprétée comme une mesure de la capacité de ce nœud à relayer une information dans le réseau.
Le projet propose de réaliser des implantations de plusieurs mesures de centralité puis d'étudier le passage à l'échelle de ces mesures, et d'étudier les résultats produits sur des graphes de réseaux réels de diverses natures (e.g. graphes de réseaux sociaux, graphes d'échanges de mails, graphes d'infrastructure de transports).

Sujet 10 : Identification de motifs structuraux dans les protéines

— Encadrant(e) : Mathilde Carpentier (`mathilde.carpentier@upmc.fr`)
— Titre : Identification de motifs structuraux dans les protéines.
— Nombre d'étudiants : 2 ou 3
— Description :
L'algorithme de Karp-Miller-Rosenberg est un algorithme de détection de répétitions dans une structure de données (chaînes de caractères, arbres, tableaux). Il est l'œuvre de Richard Karp, Raymond Miller et Arnold Rosenberg et date de 1972 [1].
La version originelle de l'algorithme KMR permet de construire rapidement des motifs de taille n par juxtaposition des motifs de taille n/2, elle est alors séquentielle. Il existe aussi une version généralisée de l'algorithme KMR [2] identifie des alignements de mots qui sont similaires mais pas nécessairement identiques et qui se produisent approximativement au même endroit dans toutes les chaînes. La méthode fonctionne en deux étapes successives. Tout d'abord, un algorithme rapide est utilisé pour établir un répertoire de motifs exactement répétés apparaissant dans une majorité donnée de chaînes. Deuxièmement, l'algorithme construit des motifs d'ancrage récursifs par une stratégie de division et de conquête et converge sur un nombre maximal d'alignements. Cette méthode pourrait être étendue à l'analyse d'images bidimensionnelles.
L'objectif de ce projet est d'adapter cet algorithme pour trouver des les contacts conservés au sein des structures protéiques. Une protéines est une chaine d'acides aminés. On dit que cette chaine se replie dans l'espace 3D. Ainsi, certains acides aminés distants dans la chaine peuvent être proche en 3D. S'ils sont suffisamment proches, deux acides aminés sont dits en contact et ils peuvent interagir. Ces interactions peuvent être indispensable pour maintenir une structure stable et il a été montré que dans les familles de protéines de structure similaire, les contacts entre acides aminés sont conservés. Les contacts entre les n acides aminés d'une protéine peuvent être représentés sous la forme d'une matrice $n \times n$ avec 1 si les 2 acides aminés sont en contact et 0 sinon. Rechercher les contacts conservés dans une famille de k protéines consiste donc à comparer k matrices. Nous voulons adapter l'algorithme de KMR à ce problème. Le programme sera préférentiellement écrit en python.

1. Karp, R., Miller, R., and Rosenberg, A. (1972). Rapid identification of repeated patterns in strings, trees and arrays. STOC '72 : Proceedings of the Fourth Annual ACM Symposium on Theory of Computing.

2. Landraud,A.M., Avril,J.-F. and Chretienne,P. (1989) An algorithm for finding a common structure shared by a family of strings. IEEE Transactions on Pattern Analysis and Machine Intelligence, 11, 890–895.

Sujet 11 : Ordonnancement de tâches malléables avec des contraintes topologiques

— Encadrant(e) : Evripidis Bampis (`evripidis.bampis@upmc.fr`)
— Titre : Ordonnancement de tâches malléables avec des contraintes topologiques.
— Nombre d'étudiants : 2 ou 3
— Description : Les besoins en puissance de calcul ont augmenté de manière significative ces dernières années. Cette puissance de calcul s'accompagne par une consommation énergétique de plus en plus grande. Afin de faire face à cette situation, la taille des plateformes de calcul haute performance a considérablement augmentée. La conception d'algorithmes de placement de tâches plus efficaces est désormais une priorité dans le domaine du calcul haute performance. Dans ce projet, nous proposons un nouveau cadre de modélisation dans lequel nous réduisons l'espace des ordonnancements réalisables via l'ajout de contraintes sur l'affectation des tâches à des machines de calcul et à des machines d'Entrées/Sorties (E/S). Plus précisément, nous considérons des tâches pouvant utiliser plusieurs machines simultanément, et nous nous intéressons à l'impact de la contrainte de contiguïté (les machines utilisées pour une tâche doivent être contigües), ainsi que celle de la localité (chaque tâche doit être adjacente à une machine d'E/S), sur la date à laquelle toutes les tâches sont terminées. Ces deux contraintes ont pour objectif la réduction du surcoût de communication dont l'impact est important non seulement sur la date de fin des tâches, mais aussi sur la consommation énergétique. Le travail consistera à implémenter des algorithmes d'ordonnancement et à les évaluer à l'aide de simulations pour des topologies (façons dont les machines sont disposées) particulières. Nous allons également intégrer dans notre modélisation la notion de la malléabilité, c'est-à-dire de la possibilité de choisir le nombre de machines utilisées pour l'exécution d'une tâche (plus le nombre de machines sera grand, plus l'exécution de la tâche sera rapide). La conception d'une interface graphique appropriée est également demandée.

Sujet 12 : Ordonnancements équitables

— Encadrant : Fanny Pascual (`Fanny.Pascual@lip6.fr`)
— Titre : Ordonnancements équitables
— Nombre d'étudiants : 2 ou 3
— Description : Les problèmes d'ordonnancement prennent comme données un ensemble de tâches, un ensemble de machines, et une fonction objectif (un but). Ils consistent à exécuter les tâches sur les machines de façon à optimiser la fonction objectif, sachant qu'une machine ne peut exécuter qu'une seule tâche à la fois. Par exemple, chaque tâche peut avoir une durée d'exécution et une deadline (date limite) à laquelle elle doit être exécutée, et le but peut être d'exécuter les tâches de manière à minimiser le nombre de tâches en retard, ou la somme des retards des tâches. Ces problèmes ont été très étudiés depuis une cinquantaine d'années, du fait de leurs nombreuses applications pratiques.
Le but de ce projet est de coder (et éventuellement concevoir) des algorithmes d'ordonnancements en présence de plusieurs utilisateurs qui chacun possèdent des tâches à ordonnancer sur des machines partagées. On s'intéressera tout d'abord à coder certains algorithmes classiques

d'ordonnancement, puis à étendre ces algorithmes au cas où il y a plusieurs utilisateurs afin de retourner des ordonnancements équitables (plusieurs notions d'équité pourront être utilisées). Une interface graphique permetant de visualiser les ordonnancements obtenus pourra être proposée.