

Sujets de projet

Introduction

Le but de ces projets est de vous initier à la programmation mathématique. Chaque sujet contient un exemple de résolution mathématique. Cela signifie que vous devez programmer des solutions à ces problèmes et (sauf mention contraire) vous ne devez pas utiliser de librairie qui implémente les solutions standards.

Tous les projets demandent de faire un programme avec un minimum de caractéristiques pour avoir la moyenne. Chacun propose une série d'améliorations possibles qui permettront d'augmenter la note. Il faut utiliser l'un des 3 langages principalement enseignés en licence : C, C++ ou Java. Ces programmes doivent pouvoir compiler et être exécutés sur les ordinateurs de l'université, et le rendu devra contenir toutes les librairies nécessaires. Ils devront contenir un moyen simple de compiler les programmes ainsi qu'une petite documentation.

1. Logiciel de dessin vectoriel

Techniques: Lecture XML , Courbe de Bézier, dessin.

**Responsable
du sujet:** Fabien Rico

Vous devez créer un logiciel capable de lire un fichier [SVG](#). C'est un langage de description de dessin vectoriel issu du web. Il se présente sous la forme d'un fichier [XML](#) que vous devrez lire en utilisant une librairie existante du langage utilisé. Ce fichier contient des descriptions d'objets géométriques (rectangles, cercles ...) mais aussi des chemins (*path*) qui permettent de définir des figures complexes notamment grâce à des [courbes de Bézier](#) cubiques et des arcs elliptique.

Votre logiciel ne doit pas forcément être capable d'afficher tous les fichiers SVG existants, mais il ne doit pas faire d'erreur lorsqu'il rencontre une commande non gérée. De plus, il doit implémenter l'ensemble des possibilités des chemins.

Pour améliorer le rendu vous pouvez de plus :

- améliorer l'interface graphique (gestion des fichiers, animation, zoom ...);
- gérer l'export d'images en différents format;
- améliorer le support du langage SVG (remplissage, lien, ...).

2. Classification par arbre de décision et forêt aléatoire

Techniques: Classification, statistiques

**Responsable
du sujet:** Fabien Rico

Info a voir:

- [Ensembles de cours datamining \(cf *Arbre de décision* et *Random Forest*\)](#)

La classification automatique est un domaine où on cherche à apprendre à reconnaître la classe d'individus à partir d'exemples. Une technique parmi les plus efficaces est basée sur la construction d'un grand nombre d'arbres de décision (ce qu'on appelle forêt aléatoire). En effet, l'expérience montre que si on construit suffisamment de classifieurs différents entre eux mais efficaces, le fait de décider en fonction de la majorité des classifieurs apporte un grand gain.

Le but de votre projet est d'implémenter un algorithme classique de construction d'une forêt.

Pour cela il faut :

- implémenter la construction d'un arbre de décision à partir d'un jeu de données d'exemples;

- implémenter un système permettant de générer un grand nombre d'arbres de décision;
- mettre en place un système de test permettant de mesurer la qualité de vos algorithmes.

Votre logiciel devra être capable à partir d'un fichier d'exemple dans un format simple (CSV) de construire un modèle. Puis à partir d'un modèle et d'un fichier de test dans le même format de trouver les classes des objets.

Les améliorations possibles de votre travail pourront porter sur les différentes optimisations ou adaptations de l'algorithme à des cas spécifiques, sur l'amélioration du temps de calcul et du coût mémoire de votre programme. Vous pourrez aussi ajouter des prétraitements pour résoudre les cas où les données sont lacunaires ou ajouter une interface graphique au programme.

3. Multi dimensionnal scalling

Techniques: calcul de valeur propres, visualisation de données

Responsable Fabien Rico

du sujet:

Information à voir: [Alain Baccini et Philippe Besse, Exploration Statistique, chapitre 7](#)

Le [multi-dimensionnal scalling](#) (ou positionnement multi-dimensionnel MDS) est une technique qu'on utilise souvent pour visualiser des données. Cela permet de calculer des coordonnées de points dont on ne connaît que les distances 2 à deux. On peut ainsi retrouver les coordonnées de points existants, mais aussi créer des cartes utilisant d'autres informations que la distance euclidienne (à vol d'oiseau). Par exemple, repositionner les villes d'un pays en fonction du temps nécessaire pour les joindre en transport en commun, positionner les députés en fonction de leurs votes ...

Votre programme devra implémenter la construction MDS de donnée utilisant l'algorithme classique qui se résume à calculer des valeurs propres d'une matrice obtenue à partir de la matrice de distances. Il devra à partir d'un fichier contenant la liste des points et leur distance, fournir un fichier contenant les coordonnées calculées de ces points.

Pour améliorer le programme, vous pourrez faire:

- une interface graphique pour manipuler les fichiers et visualiser le résultat;
- implémenter d'autres méthodes de calcul plus robustes;
- améliorer le temps de calcul et l'utilisation mémoire de votre programme
- ...

4. Compression d'image par PCA

Techniques: calcul de valeur propres, compression

Responsable Julia Sanchez

du sujet:

Information à voir:

On cherche ici à implémenter un algorithme de compression d'image avec perte (comme jpeg). L'image que l'on utilise (ou une partie de celle-ci) peut être vue comme une matrice. On peut utiliser sur cette matrice des techniques de réduction de dimension, c'est à dire obtenir une matrice de dimension plus petite qui contient l'essentielle de l'information de départ. C'est le rôle de l'analyse en composante principale (PCA). A partir de cette matrice plus petite (comprimée), on peut reconstruire l'image de départ avec quelques déformations dépendant du taux de compression utilisé.

Votre programme devra être capable de lire une image sous un format simple (sans perte), et produire un fichier comprimé. Il devra bien sûr permettre d'appliquer le traitement inverse.

Pour améliorer ce dernier vous pourrez :

- ajouter une interface graphique;
- mettre en place un système de comparaison avec la représentation jpeg;
- améliorer le temps de calcul et le cout mémoire de votre programme.

5. Algorithme de cryptographie à clef publique RSA

Techniques: Calcul sur des entiers de taille quelconque, authentification, signature

**Responsable
du sujet:** Fabien Rico

Les algorithmes de chiffrement asymétrique sont devenus très importants dans le domaine informatique. [RSA](#) est l'un des plus connus et se base sur l'utilisation de grand nombres premiers. L'intérêt de ces algorithmes est qu'ils permettent un échange d'informations chiffrées entre des entités qui ne partagent aucun secret au départ.

En effet, les algorithmes symétriques sont bien plus robustes (pour une même taille de clef), mais il nécessite avant de commencer d'échanger un secret entre les acteurs. Pour un serveur public dont le nombre de clients est très important, il n'est pas possible de mettre en place un canal pour échanger ce secret autrement qu'en utilisant le réseau internet. On utilise donc les algorithmes à clef publique pour sécuriser les premiers échanges et mettre en place une session sécurisée.

Vous devez implémenter l'algorithme RSA, c'est à dire :

- générer la clef;
- chiffrer avec la clef publique et déchiffrer avec la clef secrète;
- chiffrer avec la clef secrète et déchiffrer avec la clef publique (pour signer).

Comme amélioration vous pourrez :

- implémenter d'autres algorithmes de chiffrement;
- créer un petit logiciel de connection à distance sécurisé.

6. Algorithme du recuit simulé

Techniques: Coloration de graphes, métaheuristique

**Responsable
du sujet:** Aurélie Kong Win Chang

Les graphes sont un objet mathématique très utilisé. Un problème fréquemment posé consiste en trouver une coloration de graphe attribuant à chaque sommet du graphe une couleur sans que jamais deux sommets adjacents soient de la même couleur. Ceci peut se résoudre très simplement. Mais si on souhaite obtenir la même coloration en utilisant le plus petit nombre de couleurs possibles, la situation est différente. En effet, ce problème d'optimisation est NP-difficile (Il n'y a sans doute pas de solution efficace pour lui). Nous proposons donc ici de chercher non pas la meilleure solution, mais une approximation, à l'aide d'un algorithme inspiré d'une technique tirée de la métallurgie : [le recuit simulé](#).

Le projet consistera en l'implémentation d'une structure permettant de représenter un graphe et sa coloration, un algorithme simple permettant de trouver une coloration quelconque valide, et l'algorithme de recuit simulé pour trouver une approximation de la solution. Votre programme devra être capable de trouver une approximation acceptable de la solution du problème.

Améliorations possibles :

- implémenter une autre méthode pour trouver la solution optimale ou une approximation et comparer les temps de calcul et la qualité de la solution;

- ajouter une interface graphique permettant de visualiser les évolution de l'algorithme et son résultat.

7. Cueillette exploratoire

Techniques: Informatique bio-inspiré

Responsable du sujet: Aurélie Kong Win Chang

Information à voir: [Un article sur le sujet](#)

Remarque: Sujet issu du cours de M2 : informatique bio inspirée.

L'informatique bio-inspirée consiste à se baser sur des observations tirées de la nature pour, entre autres, créer de nouveaux algorithmes. Ainsi une stratégie d'exploration observée chez de nombreux animaux (albatros, humains chasseurs-cueilleurs...) a-t-elle été reprise pour des robots sous-marins. Ce projet propose d'implémenter différentes stratégies de recherche de ressources et de les comparer pour différentes configurations :

- implémenter un petit système multi-agent très simple (un terrain, des patches d'intérêt, des agents qui se déplacent) fonctionnant de manière séquentielle;
- implémenter pour ces agents une stratégie d'exploration aléatoire et une exploration suivant le vol de Lévy (en cherchant, dans ce dernier cas, les valeurs des paramètres pour lesquelles l'exploration est la plus efficace) ; le but étant de ramasser au plus vite tous les patches de nourriture;
- comparer les résultats pour différentes configurations données (fréquence et tailles des patches, nombre d'agents).

Améliorations possibles :

- il est possible de trouver les valeurs des paramètres de manière totalement expérimentales, en tâtonnant, mais d'autres méthodes sont utilisables
- proposer d'autres stratégies d'exploration et les comparer à celles déjà implémentées

8. Word embedding (plongement lexical)

Langage: C, C++, Java

Technique: analyse de text, manipulation de grandes matrices

Responsable du sujet: Fabien Rico

Info à voir: [Global Vectors for Word Representation](#) Article expliquant l'algorithme

Le word embedding est une méthode à la base de nombreux travaux d'analyse du langage naturel. Il consiste à vouloir transformer les mots de la langue en un ensemble de vecteur mathématiques. Le but est de le faire en conservant une certaine notion du sens des mot.

Le moyen de faire cela est de travailler à partir d'une matrice de co-occurences des mots dans les phrases trouvées dans un grand corpus de texte (par exemple wikipedia). Cette matrice contient une partie du sens des mots car le sens dépend du contexte. Mais cette matrice est trop grande pour être utilisée. Il faut alors appliquer une technique de réduction de dimension de la matrice pour obtenir les vecteurs souhaités.

C'est un projet plutôt exploratoire, vous allez devoir faire cette opération : lecture des textes; création de la matrice; réduction de la dimension... Ce projet n'étant pas assez maitrisé, il n'y a pas de minimum proposé. On notera en fonction de l'état d'avancement final.