

Examen ' Conduite de projets en science des données et Predictive Analytics'

- Lorsque vous utilisez une réponse trouvée sur internet, donner l'hyperlien des sources utilisées.

<<Von Campe, Hugo>>

1: IA

Décrire la différence entre approche réaliste et approche utilitariste *dans la démarche scientifique*.

Une approche réaliste est Platonienne, c'est-à-dire que le but est de comprendre le monde à partir d'observations empiriques et de le formaliser. De ce fait, le but de la science est de créer de la connaissance, d'explorer le monde des idées. L'approche utilitariste quant à elle a pour postulat que la science doit servir à quelque chose, peu importe les idées et les concepts derrière car il faut seulement que cela fonctionne. C'est une approche plus appliquée. Les deux ont pour objectif la progression dans la connaissance.

2: programmation logique/chainage avant

Dans un langage à base de règles simple en chaînage avant (on appelle cela un système de production) on a le programme:

```
var input=[] , result=[] , i=1 , tmp=0;
```

```
when input.length>0 and i >= input.length then result.append(input[tmp]),
```

```
input.removeAt(tmp), i=1, tmp=0;
```

```
when input[i] < input[tmp] then tmp=i, i=i+1;
```

```
when input[i] >= input[tmp] then i=i+1;
```

Lorsque l'instruction `input=[2,0,5,4,9]`; est exécutée, que contiendront les variables *result* et *input* en retour? Expliquer l'algorithme. Donner sa complexité.

On aurait alors `result = [0 ;2 ;4 ;5 ;9]` et `input = []`.

L'algorithme parcourt la liste et cherche la plus petite valeur, l'ajoute à *result*, et l'enlève du tableau *input*. Elle recommence ensuite jusqu'à ce qu'elle soit vide.

C'est un algorithme de tri avec une complexité linéaire (parcours de liste).

3: Smart City: Trouver sur internet 3 logiciels commerciaux professionnels destinés à remplir un rôle similaire à celui du projet SmartDeliveries. En vous basant sur la présentation commerciale, identifiez leurs principales caractéristiques démarquantes

(quelles fonctionnalités mettent-ils particulièrement en avant par rapport à la concurrence). Fournir les références utilisées.

Basé sur le cours, puis sur internet, j'ai pu trouver trois logiciels avec un rôle similaire à celui de SmartDeliveries. On peut noter :

- Mapotempo : ils mettent en avant la complétude de leur solution, 100% en ligne, une expérience utilisateur intuitives, une open source et les vues embarquables.
<https://www.mapotempo.com/>
- AntsRoute : mettent en avant l'optimisation qu'ils offrent, moins d'effort pour plus de productivité et la fidélisation de leurs clients. <https://antsroute.com/>
- Descartes : la gestion efficace de leur flotte.
<https://www.descartes.com/fr/solutions/optimisation-de-tournees-mobilite-telematique/planification-optimisation-de-tournees>

4: trafic routier

a- Quelles sont les principales variables mesurées par un détecteur de trafic?

L'id du point de détecteur, la date et l'heure, le débit par heure du nombre de véhicules, le taux d'occupation (lié à la vitesse).

La densité du trafic et la congestion (3 types)

- Liée à un événement connu ou anticipé (alertes trafic)
- Liée à un accident
- Liée à la saturation, chronique ou non du réseau

b- Qu'est ce que le diagramme fondamental d'un détecteur de trafic, pourquoi est-il utile pour mesurer et prévoir la congestion?

Le diagramme fondamental est produit en mettant en abscisses le taux d'occupation, ou la vitesse, et en ordonnée le nombre de véhicules qui passent. On constate que lorsque le TO est faible, plus y'a de véhicule qui rentre, plus y'a de véhicules qui sortent (correspond au vert sur Waze). Donc le flux, le débit augmente quand on rajoute des véhicules. Puis il arrive un seuil (correspondant à l'orange sur Waze), un certain plateau, où plus on ajoute de véhicules, plus ces véhicules vont lentement car trop près les uns des autres. Ce seuil correspond au débit maximal admissible sur le détecteur. Ensuite quand le taux d'occupation augmente, les véhicules sont proches les uns des autres, mais ils n'arrivent pas à aller plus vite (correspond au rouge sur Waze). Donc le débit n'augmente pas. Donc tout détecteur aura cette forme de courbe en cloche.

Logiquement grâce à tout ça on peut donc prévoir des patterns et prévoir la congestion à tel horaire, car finalement sur le second graphique tout ce qui est en dessous indique un débit fluide et tout ce qui est au-dessus c'est congestionné.

c- quel est le débit typique maximal d'un tronçon à une voie

- en zone urbaine

- sur voie rapide ou autoroute
- Pourquoi cette différence?

Cette différence est due fait que c'est chaotique, les relations entre les arcs et le graph, donc il y a des mouvements tournants (virages) bien plus nombreux.

5: temps de parcours

- a) **Quelles sont les principales variables prédictives du temps de parcours d'un camion de livraison en ville, par ordre d'importance décroissante ? (Déterminées en cours)**

- Distance
- Identifiants de la tournée (roundId)
- prévision du temps de trajet
- Taux d'occupation
- angle

- b) **Citer 2 facteurs potentiels affectant les temps de parcours et difficiles à mesurer avec les données fournies dans les fichiers fournis en TP.**

On peut avoir les « circonstances locales » comme les différents aléas possibles, et le taux d'erreur d'adresse (adresses incomplètes par exemple).

6. Prescriptive Analytics

Les affirmations suivantes sont-elles vraies ou fausses:

- a- **Un problème de décision est dans la classe de complexité NP si et seulement si il n'existe pas d'algorithme polynomial pour le résoudre.**

Faux, car un problème de décision appartient à la classe P (inclue dans NP) si toutes ses instances peuvent être résolues avec un algorithme polynomial.

- b- **Dans l'industrie, la majorité des problèmes d'ordonnancement sont résolus grâce à des heuristiques.**

Faux, les COPs appartiennent à la classe NP et on peut les résoudre avec de la programmation par contraintes, SAT ou Integer Linear Programming.

- c- **Le problème suivant possède exactement trois solutions:**

$u \in \{1,3\}$
 $v \in \{1,2\}$
 $w \in \{3,4\}$
 $x \in \{1,5\}$
 $y \in \{4,5\}$
allDifferent(u,v,w,x,y)

Faux, on se retrouve avec deux solutions :

- 1,2,3,5,4
- 3,2,4,1,5

d- L'algorithme de résolution de CP-Optimizer est un algorithme exact: si un problème d'optimisation est faisable, il garantit de trouver une solution optimale.

Vrai.

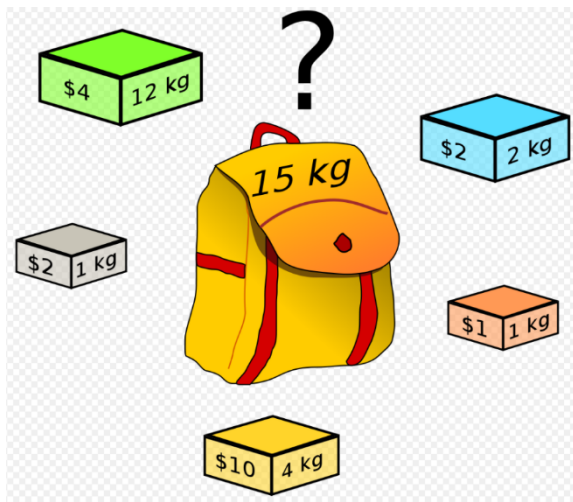
7. optimisation

a- En cherchant sur internet, décrivez un problème d'optimisation combinatoire non vu dans le cours dont la version de décision est un problème NP-Complet.

Le problème d'optimisation combinatoire que j'ai choisi est le problème du sac à dos.

https://fr.wikipedia.org/wiki/Probl%C3%A8me_du_sac_%C3%A0_dos

b- Décrivez une petite instance particulière de ce problème d'optimisation (avec des valeurs pour chacune des données).



L'instance que j'ai décidé de décrire est celle de cette image tirée de wikipédia. C'est la version avec le sac pouvant contenir maximum 15kg, et avec des poids de

- 12 kg pour 4 \$
- 1 kg pour 2 \$
- 4 kg pour 10 \$
- 1 kg pour 1 \$
- 2 kg pour 2 \$

c- Donnez une solution faisable non-optimale et une solution optimale de cette petite instance.

En réalisant l'arbre de décision, nous pouvons identifier la solution faisable optimale qui comprend :

- Le poids de 1 kg pour 2 \$
- Le poids de 4 kg pour 10 \$
- Le poids de 1 kg pour 1 \$
- Le poids de 2 kg pour 2 \$

On obtient donc 15 \$ pour une somme de poids de 8 kg, inférieure à 15 kg.

Une solution faisable non-optimale est la suivante :

- Le poids de 1 kg pour 2 \$
- Le poids de 4 kg pour 10 \$
- Le poids de 2 kg pour 2 \$

On obtient donc une somme de 14 \$ pour 7 kg.

8. programmation par contraintes

Deux principes fondamentaux de la Programmation par Contraintes sont (1) la recherche arborescente et (2) le filtrage du domaine des variables. Décrivez brièvement ces principes, leurs rôles et la façon dont ils sont mis en œuvre durant la résolution.

Le principe de la recherche arborescente est l'exploration d'un arbre de recherche. Par exemple sur le sudoku, on peut identifier dans un premier temps pour chaque case chaque possibilité jusqu'à trouver la bonne solution, et continuer de suite pour toutes les cases. On va donc explorer l'arbre en essayant de fixer une par une les variables de décision. A partir de la racine du problème (représentation initiale), je cherche à fixer la valeur d'une case (sudoku). On part d'une approche où il y'a tout le domaine des variables initiales (1 à 9 pour les cases de sudoku) et on fait la propagation. On va donc tester un ensemble de possibilités pour arriver à la solution (avec 9x9 solutions à la fin).

Le filtrage du domaine des variables filtre le domaine des valeurs possibles. Un algorithme de filtrage est associé à chaque contrainte. Son rôle va alors être de supprimer les valeurs impossibles dues à la contrainte. Le but est d'essayer de la satisfaire, et de réduire la combinatoire d'un problème en retirant les valeurs non pertinentes. C'est typiquement la façon dont l'on résonne pour résoudre un sudoku, les contraintes étant les chiffres présents sur les lignes et les colonnes et dans les cases : on élimine ces possibilités pour trouver la bonne solution.