

MATH80629 – Apprentissage automatique I : analyse des mégadonnées et prise de décision
Examen. Décembre 2020.

Professeur: Laurent Charlin

Il y a 4 questions dans cet examen, pour un total de 84 points.

Question:	1	2	3	4	Total
Points:	17	16	22	29	84
Score:					

Veillez répondre dans les fichiers gabarits fournis. Quand c'est nécessaire, vous pouvez ajouter des figures dans ces gabarits (vous pouvez simplement dessiner à la main et en prendre une photo ou bien utiliser un programme de dessin.)
L'espace vide sous chaque question de ce questionnaire est une indication de la longueur maximale souhaitable de votre réponse. Il est possible que vos réponses soient bien plus courtes.

1. [17 points]

(a) Indiquez auquel des trois paradigmes chaque situation (i, ii, iii, iv) correspond. Paradigmes :

- a) apprentissage supervisé (supervised learning)
- b) apprentissage non-supervisé (unsupervised learning)
- c) apprentissage par renforcement (reinforcement learning).

Veillez aussi brièvement justifier votre réponse (maximum 3 lignes).

i. (2 points) Vous avez un jeu de données de haute dimension qui décrivent la position de molécules dans un système complexe. Vous voulez explorer les données en les visualisant dans un espace de basse dimension (par exemple, en 2 dimensions).

ii. (2 points) Vous devez développer un système pour traduire du texte de l'anglais au français. Vous avez d'ailleurs accès à un ensemble de texte en anglais et à leur traduction française.

- iii. (2 points) Vous devez programmer un système pour jouer à un jeu vidéo. Le jeu implique un personnage qui doit en sauver un autre. Le personnage principal doit séquentiellement passer à travers des tableaux (des niveaux). Pour chaque tableau, le personnage doit effectuer une séquence d'actions pour aller au prochain tableau.

- iv. (2 points) Qu'en est-il si dans l'exemple précédent vous voulez plutôt obtenir un système pour reconnaître les objets à partir de l'écran du jeu vidéo pendant l'entièreté du jeu?
-

(b) (2 points) Décrivez une situation (une tâche) où une combinaison d'apprentissage supervisé et non-supervisé serait utile?

(c) (4 points) Qu'est-ce que la régularisation (*regularization*). Veuillez en donner un exemple.

- (d) (3 points) Qu'est-ce que l'hypothèse i.i.d (*i.i.d. assumption*). Décrivez aussi pourquoi on en a besoin en apprentissage automatique.
-

2. [16 points]

Apprentissage Non-supervisé

- (a) (4 points) Pour le regroupement (*clustering*), donnez deux avantages d'utiliser une mixture de Gaussiennes (*Gaussian mixture model GMM*) plutôt qu'un modèle K-moyennes (*K-Means*)?

- (b) (6 points) K-moyennes (*K-Means*) peut seulement regrouper des données ayant des valeurs réelles (*real-value data*). Comment modifieriez-vous l'algorithme K-moyennes pour regrouper d'autres types de valeurs (par exemple, des données catégorielles ou entières)? L'algorithme est donné plus bas. Vous pouvez utiliser le numéro associé à chaque ligne pour expliquer vos changements.

```

1 Algorithm: K-Means clustering
2 Inputs: Number of clusters  $K$  and data  $x_1, x_2, \dots, x_n$ ;
3 Initialize cluster centers  $\mu_1, \mu_2, \dots, \mu_K$ ;
4 Initialize cluster responsibilities  $r_1, r_2, \dots, r_n$ ;
5 while Not converged do
6   1) Update cluster responsibilities:
7      $r_i = \arg \min_{r_i} \sum_k r_i \|x_i - \mu_k\|^2 \quad \forall i$ ;
8   2) Update cluster centers:
9      $\mu_k = \arg \min_{\mu_k} \sum_i r_i \|x_i - \mu_k\|^2 \quad \forall k$ ;
10 end

```

- (c) (6 points) Pourriez-vous utiliser Apache Spark pour distribuer l'algorithme K-Moyennes? Si oui, veuillez décrire comment faire et quel gain potentiel (par exemple, en termes de temps de calcul) vous pourriez obtenir. Sinon, veuillez justifier votre réponse.

3. [22 points]

Apprentissage Profond

(a) Taille d'un réseau de neurones.

i. (3 points) Quel est le nombre minimal de paramètres d'un réseau de neurones complètement connecté (*fully-connected feed-forward*) pour modéliser des données avec 100 attributs (*features*) et deux sorties (*outputs*)?

ii. (3 points) Pour la même tâche, combien a de paramètres un réseau avec deux couches cachées de 20 unités chacune (*2 hidden layers of size 20 each*)?

- (b) Réseau de neurones pour les systèmes de recommandations.
- i. (8 points) Décrivez une façon de modéliser des notes (*ratings*) d'utilisateurs pour des items avec un réseau de neurones. Vous pouvez imaginer que vous avez accès à des notes précédentes. Chaque donnée est un tuple contenant l'identité d'un utilisateur, l'identité d'un item, et une note. Chaque note est un chiffre entre 0 et 1.

$$X = [(1, 5), (10, 2), (2, 10), \dots, (5, 5)]$$

$$Y = [0.1, 0.5, 0.2, \dots, 0.9]$$

Ces données montrent notamment que l'utilisateur 1 a donné une note de 0.1 à l'item 5. Soyez aussi précis que possible. Vous pouvez répondre avec une esquisse (*sketch*) du réseau proposé.

- ii. (8 points) Imaginez une autre situation où plutôt que de modéliser les notes, vous aimeriez modéliser la séquence des items consommés par un utilisateur. En plus, vous imaginez que les derniers items consommés sont indicateurs des prochains items à consommer. Suggérez un second réseau de neurones à utiliser dans cette situation. Vous êtes de nouveau encouragé à répondre en utilisant un dessin de votre réseau de neurones. Spécifiez clairement : 1) ses entrées, 2) ses sorties et 3) comment le modèle sera utilisé pour recommander des items à des utilisateurs (c'est-à-dire au moment du test).

4. [29 points]

Apprentissage par renforcement

(a) (2 points) Décrivez l'utilité du facteur de dévaluation (*discount factor*) γ dans un processus de décision Markovien (*Markov decision process MDP*).

(b) Pour les prochaines sous questions prenez en compte la méthode *Monte Carlo with exploring starts* ci-bas (recopiée des diapos du cours).

```

Monte Carlo ES (Exploring Starts)
Initialize, for all  $s \in \mathcal{S}$ ,  $a \in \mathcal{A}(s)$ :
   $Q(s, a) \leftarrow$  arbitrary
   $\pi(s) \leftarrow$  arbitrary
   $Returns(s, a) \leftarrow$  empty list

Repeat forever:
  Choose  $S_0 \in \mathcal{S}$  and  $A_0 \in \mathcal{A}(S_0)$  s.t. all pairs have probability  $> 0$ 
  Generate an episode starting from  $S_0, A_0$ , following  $\pi$ 
  For each pair  $s, a$  appearing in the episode:
     $G \leftarrow$  the return that follows the first occurrence of  $s, a$ 
    Append  $G$  to  $Returns(s, a)$ 
     $Q(s, a) \leftarrow \text{average}(Returns(s, a))$ 
  For each  $s$  in the episode:
     $\pi(s) \leftarrow \text{argmax}_a Q(s, a)$ 

```

i. (5 points) Quel est le but de cette méthode?

ii. (3 points) Qu'entend-on par "exploring starts"? Dans les environnements où cette technique n'est pas pratique, quelle autre technique peut-on utiliser à sa place?

iii. (10 points) Imaginez un environnement avec 4 états (A, B, C, D) et 2 actions (L, R). Pour tous les états et actions, obtenez $Q(s, a)$ ainsi que $\pi(s)$ calculées par la méthode Monte Carlo ES après deux épisodes.
Les éléments de $Q(s, a)$ ont été initialisés à 0.0. $\pi(s)$ a été initialisée uniformément et aléatoirement.
De plus, $\gamma = 1$.

Épisode 1 : $(B, L, 5) \rightarrow (A, L, 5) \rightarrow (C, L, 2) \rightarrow (A, L, 5) \rightarrow (C, R, 0)$;
Épisode 2 : $(C, R, 2) \rightarrow (A, L, 5) \rightarrow (C, L, 2)$;

Chaque tuple représente un état, une action et une récompense. Par exemple, (A,L,5) voudrait dire que l'agent a démarré dans l'état A, a exécuté l'action L et a reçu une récompense de 5.

(c) Vous devez créer un agent d'apprentissage par renforcement pour conduire une voiture (conduite autonome). Décrivez exactement comment modéliser les actions, les états et les récompenses.

i. (2 points) Actions :

ii. (4 points) États (*States*) :

iii. (3 points) Récompenses (*Rewards*) :