

## Examen - Session 2

Deux feuilles A4 autorisées, calculatrice autorisée, barème indicatif

### Exercice 1 (5 points) – Questions indépendantes

**Q 1.1** Soit un réseau de neurones à deux entrées, une couche cachée de deux neurones et un neurone de sortie. On suppose toutes les fonctions d'activation linéaire.

**Q 1.1.1** Dessinez ce réseau en précisant tous les poids du réseau.

**Q 1.1.2** Est-il possible de représenter ce réseau par un perceptron? Si oui, donner son équivalent.

**Q 1.2** Répondre par vrai ou faux **en justifiant** votre réponse en une ligne :

1. La regression logistique apprend des frontières non-linéaires car la fonction logistique est non linéaire.
2. Sur des données séparables linéairement, l'algorithme du perceptron converge vers des minima locaux du fait de l'initialisation aléatoire des poids.
3. Les arbres de décision sont plus expressifs que les  $k$ -plus proches voisins.
4. L'estimation de densité est un préalable à tout algorithme de classification.

**Q 1.3** Un étudiant décide d'utiliser de l'apprentissage par renforcement pour jouer au loto (6 numéros à trouver dans l'ordre parmi 49). A-t-il raison et que peut-il espérer ?

**Q 1.4** Donner un réseau de neurones capable de classifier le plan 2D en deux classes : en positif l'ensemble des points dans le carré centré en  $(0, 0)$  et de coté de longueur 2, en négatif tous les autres points du plan.

**Q 1.5** Rappel : un noyau, fonction  $K : X \times X \rightarrow \mathbb{R}$  est dit admissible s'il existe une fonction  $\phi : X \rightarrow \mathbb{R}$  (fonction de projection ou *feature map*) et  $K(x, x') = \langle \phi(x), \phi(x') \rangle$ . Montrez que si  $K$  et  $K'$  sont deux noyaux que :

**Q 1.5.1**  $cK$  est un noyau pour  $c \in \mathbb{R}^+$

**Q 1.5.2**  $K + K'$  est un noyau.

### Exercice 2 (6 points) – Apprentissage d'ordonnement

L'objectif du *ranking* (ou ordonnancement) est de prédire des préférences sur les items considérés. L'exemple classique de machines d'ordonnement sont les moteurs de recherche (tels que Google par exemple) qui retournent des listes ordonnées de résultats en réponse à un besoin d'information. La tâche d'un moteur de recherche est alors de décider quel document placer au dessus de quel autre document.

Soient  $N$  échantillons étiquetés  $X = \{\mathbf{x}^i\}_{i=1, \dots, N}$  codés sous forme de vecteurs à  $d$  dimensions :  $\mathbf{x}^i \in \mathbb{R}^d$ . Les étiquettes sont réelles :  $Y = \{y^i\}_{i=1, \dots, N}, y^i \in \mathbb{R}$ . Plutôt que d'apprendre directement à associer les étiquettes aux vecteurs d'entrée, ce qui peut dans certains cas se révéler une tâche difficile, on se contente de respecter des ordres de préférence sur les items : plus  $y_i$  est grand, plus il doit être préféré selon notre prédicteur  $f(\mathbf{x}^i)$ .

Nous proposons d'optimiser la fonction coût suivante :

$$L = \sum_{(i,j), y^i > y^j} (y^i - y^j) \max(0, \gamma - (f(\mathbf{x}^i) - f(\mathbf{x}^j))), \text{ avec } \gamma = 1$$

Dans toute la suite, nous proposons d'utiliser un prédicteur linéaire :

$$f(x^i) = \sum_j x_j^i w_j = \langle \mathbf{x}^i, \mathbf{w} \rangle \in \mathbb{R} \text{ (produit scalaire classique)}$$

**Q 2.1** Interprétation de la fonction de coût.

**Q 2.1.1** A quoi correspond une valeur nulle du coût pour une paire d'échantillons? une valeur positive de coût? Comment se comporte le coût en fonction de l'écart entre  $y_i$  et  $y_j$ ?

**Q 2.1.2** Que se passe-t-il si on augmente la valeur de  $\gamma$ ? Et pour une valeur nulle de  $\gamma$ ?

**Q 2.2** Optimisation : nous utiliserons une technique de gradient stochastique. Le gradient est calculé à chaque étape sur seulement une paire de points tirés au hasard.

**Q 2.2.1** Calculer le gradient du coût par rapport à  $\mathbf{w}$  pour un couple  $(i, j)$ .

**Q 2.2.2** Rappeler la définition d'un algorithme de descente de gradient (cas général). Donner ensuite l'algorithme pour ce cas particulier. Rappeler les entrées (hyper-paramètres de la méthode) et les sorties de cet algorithme.

**Q 2.2.3** Cet algorithme étant itératif, proposer au moins deux critères d'arrêt en justifiant succinctement les points forts et faibles de ces critères (en terme de facilité d'implémentation ou de coût de calcul par exemple).

**Q 2.3** Preuve de bon fonctionnement. Soit  $y^i > y^j$  :

**Q 2.3.1** Exprimer  $\mathbf{w}^{t+1}$  en fonction de  $\mathbf{w}^t$  lorsqu'il y a mise-à-jour (application de la descente de gradient sur une itération en considérant que l'on a choisi un exemple mal classé)

**Q 2.3.2** Montrer que  $f^{t+1}(\mathbf{x}^i) - f^{t+1}(\mathbf{x}^j) = f^t(\mathbf{x}^i) - f^t(\mathbf{x}^j) - \epsilon(y^i - y^j)\|\mathbf{x}^i - \mathbf{x}^j\|^2$ , avec  $\epsilon$  le pas de gradient

**Q 2.3.3** Comparer  $f^{t+1}(\mathbf{x}^i) - f^{t+1}(\mathbf{x}^j)$  et  $f^t(\mathbf{x}^i) - f^t(\mathbf{x}^j)$ . Le résultat obtenu vous semble-t-il logique? Expliquer rapidement pourquoi.

### Exercice 3 (6 points) – Filtrage collaboratif

Le but de l'exercice est de proposer un modèle pour la recommandation (ou filtrage collaboratif) : il s'agit de prédire l'avis d'un utilisateur sur un item (qui peut être un film, un produit, un restaurant, ...) en fonction de votes connus entre  $n$  utilisateurs (ensemble  $\mathcal{U}$ ) et  $m$  items (un ensemble  $\mathcal{V}$ ). Les votes (ou scores) que les utilisateurs ont donné aux items sont représentés par une matrice de votes (ratings)  $R$  de taille  $n \times m$ , avec  $r_{ij}$  le vote de l'utilisateur  $i$  pour l'item  $j$ . On suppose tous les votes strictement positif (par exemple entre 1 et 5), une valeur nulle pour  $r_{ij}$  signifiera donc que le score de l'utilisateur  $i$  pour l'item  $j$  n'est pas connu.

Une modélisation courante de ce contexte est de considérer un espace *latent* - non explicite mais exprimé dans les données - de facteurs explicatifs de l'avis des utilisateurs sur les items. Par exemple, dans le cas de la recommandation de films, on peut imaginer que cet espace latent est composé des différents genres de films (comédie, thriller, romance, ...). Un film peut être alors représenté par un ensemble de poids sur les différents genres - son vecteur de poids - par affinité de chaque genre avec le film. De même, un utilisateur peut être représenté par son vecteur de poids sur les différents genres. Le score prédit pour un utilisateur sur un film est alors le produit scalaire entre les deux vecteurs de poids. L'erreur au moindres carrés entre le score prédit et le score observé est généralement utilisée comme fonction de coût.

**Q 3.1** On suppose un espace latent de dimension 4 : (comédie, romance, thriller, policier) et les données suivantes :

	comédie	romance	thriller	policier
<b>films</b>				
$v_1$ : Angry Bird	0.4	0.2	0.	0.
$v_2$ : The Big Lebowski	0.8	0.2	0.6	0.4
$v_3$ : Fargo	0.4	0.1	0.3	0.2
$v_4$ : Saw	0	0	0.6	0.1
<b>users</b>				
$u_1$	0.5	0.5	0.5	0.5
$u_2$	0.2	0.2	0.6	0.8
$u_3$	0.1	0.1	0.3	0.8

user	film	score
$u_1$	$v_1$	3
$u_1$	$v_3$	10
$u_2$	$v_2$	9
$u_2$	$v_3$	4
$u_3$	$v_3$	3
$u_3$	$v_4$	3

**Q 3.1.1** Calculer la matrice de scores  $R$  pour ces données.

**Q 3.1.2** Comparer les scores obtenus avec les scores de supervision, donner le coût de cette approximation. Quel opération élémentaire permet d'améliorer grandement le coût ? Est-ce important ?

**Q 3.1.3** Que remarquez vous pour les scores et la représentation des films  $v_2$  et  $v_3$  ? Sont-ils proches ou éloignés ? Expliquez le phénomène observé en tenant compte de la moyenne des votes pour chaque film. La distance euclidienne vous semble-t-elle adaptée pour calculer la proximité entre films (resp. entre utilisateurs) ? Quelle autre distance utilisée ?

**Q 3.2** Soit la matrice des scores  $R$  de taille  $n \times m$ , chaque ligne  $r_i$  correspondant aux scores donnés par un utilisateur et chaque colonne aux scores reçus  $r_j$  par un item. On considère que toute la matrice est connue. Formaliser le problème de recommandation décrit dans l'énoncé comme la recherche de deux matrices  $U$  et  $V$  de tailles  $n \times d$  et  $d \times m$  qui minimisent une expression faisant intervenir  $R$  également. A quoi correspondent  $U$ ,  $V$  et  $d$  ?

**Q 3.3** Au terme trouvé à la question précédente, on ajoute le terme  $\lambda(\sum_i \|u_{i,\cdot}\|^2 + \sum_j \|v_{\cdot,j}\|^2)$  pour établir la quantité finale à optimiser. Que représente les sommes de ce second terme ? Quel intérêt ?

**Q 3.4** En réalité, nous disposons rarement de tous les scores : la matrice  $R$  est parcimonieuse, avec très peu de valeurs connues. Comment adapter le problème d'optimisation ?

**Q 3.5** Le problème n'est pas directement optimisable : pourquoi ? Une solution est d'optimiser par alternance les deux matrices  $U$  et  $V$  : à un pas de temps,  $U$  est optimisée en considérant  $V$  fixée, à celui d'après  $V$  est optimisée en considérant  $U$  fixée.

**Q 3.5.1** A quelle famille classique de problèmes d'apprentissage statistique correspond chacune des deux phases d'optimisation ?

**Q 3.5.2** Pour un rating  $r_{i,j}$ , donner les formules de mise-à-jour associées. Donner l'algorithme complet d'optimisation (en considérant une optimisation stochastique).

**Q 3.5.3** Donner l'implémentation python.

**Q 3.6** Tous les utilisateurs n'ont pas la même moyenne d'avis (un utilisateur peut être plus sévère qu'un autre mais avoir les mêmes goûts), de même pour les items. Que manque-t-il au modèle pour pouvoir représenter ces biais ? Donner les modifications à apporter à l'algorithme.

---

**Exercice 4 (4 points) – Risque**

---

Soit le résultat d'une analyse médicale exprimé par un réel  $x \in \mathbb{R}$ , soit deux classes  $y_-, y_+$  pas malade

et malade. On sait que  $P(y = y_+ | x) = \begin{cases} 0 & \text{si } x \leq 0 \\ x & \text{si } 0 \leq x \leq 1. \\ 1 & \text{si } x \geq 1 \end{cases}$ .

On utilise un classifieur de type  $f_\theta(x) = \begin{cases} y_+ & \text{si } x > \theta \\ y_- & \text{si } x \leq \theta \end{cases}$ .

**Q 4.1** Définir le coût 0-1 sur ce problème (en français et de manière formelle). Calculer le classifieur optimal pour ce coût.

**Q 4.2** On sait qu'il est 3 fois plus couteux de classer un malade en pas malade que l'inverse.

**Q 4.2.1** Donner une fonction de coût associée et la formulation du risque.

**Q 4.2.2** Quel est le classifieur optimal dans ce cas ?

**Q 4.2.3** On suppose le résultat du test  $x$  uniformément répartie dans  $[-1, 1]$ . Quelle est la valeur du risque minimale ?

**Q 4.3** Pourrait-on faire mieux avec un classifieur bayésien ? Un classifieur bayésien naïf ?