TD 9

Exercice 1 – RKHS

Rappels : Un espace vectoriel E sur $\mathbb{R}$ est un ensemble d'éléments tel qu'il est possible de faire des combinaisons linéaires de ses éléments (E est muni d'une opération d'addition et d'une opération de multiplication par un scalaire) ;
Une fonction $Q : E \times E \to \mathbb{R}$ est un produit scalaire ssi :

1. elle est symétrique : $Q(x, y) = Q(y, x)$;
2. elle est bilinéaire : $Q(\lambda_1 x_1 + \lambda_2 x_2, y) = \lambda_1 Q(x_1, y) + \lambda_2 Q(x_2, y)$;
3. elle est positive $Q(x, x) \geq 0$ et $Q(x, x) = 0 \iff x = 0$

On notera souvent $Q(x, y) =< x, y >_E$ et la norme d'un produit scalaire $\|x\|_Q = \sqrt{Q(x, x)}$; Un espace de Hilbert est est un espace vectoriel complet muni d'un produit scalaire ; Un noyau est une fonction $k : X \times X \to \mathbb{R}$ tel qu'il existe un espace de Hilbert $\mathcal{H}$ et une fonction (de projection ou *feature map*) $\phi : X \to \mathcal{H}$ telle que $\forall x, x' \in X, k(x, x') =< \phi(x), \phi(x') >_{\mathcal{H}}$
Soit $x_1, \cdots, x_n \in X$, une fonction $k : X \times X \to \mathbb{R}$, la matrice de Gram de K est la matrice $\mathcal{K} := k_{i,j} = k(x_i, x_j)$. Une matrice est dite définie semi-positive si $\forall c_i \in \mathbb{R}, \sum_{i,j} c_i c_j k_{i,j} \geq 0$. Dans ce cas, la fonction est dite également définie positive.

Q 1.1 Exprimez $\sum_{i,j} c_i c_j k_{i,j}$ par un produit scalaire. Montrez qu'un noyau est défini positif.

Q 1.2 Le but de cette question est de montrer la contraposée, qu'une fonction symétrique semi définie positive $k : X \times X \to \mathbb{R}$ est un noyau. Pour cela, il nous faut trouver un espace hilbertien $\mathcal{H}$, un produit scalaire $Q : \mathcal{H} \times \mathcal{H} \to \mathbb{R}$ et une projection $\phi : X \to \mathcal{H}$ telle que $k(x, y) = Q(\phi(x), \phi(y))\ \ \forall x, y \in X$. On va considérer $\mathcal{H}$ l'espace vectoriel engendré par les fonctions de la forme $y \to k(y, x)$ pour tout $x \in X$. Un élément de $\mathcal{H}$ est donc une fonction de $X \to \mathbb{R}$.
Soit $\Phi : X \to \mathcal{H} := k(., x)$ un mapping de X aux fonctions de $\mathcal{H}$, $\Phi(x)(x') = k(x', x)$. Soient $\alpha_i \in \mathbb{R}$, $\beta_i \in \mathbb{R}$, $x_i \in X$, $x'_i \in X$ pour $i \in \{1..n\}$. On définit :

$$f(.) = \sum_{i=1}^{n} \alpha_i \Phi(x_i)(.), \quad g(.) = \sum_{i=1}^{n} \beta_i \Phi(x'_i)(.), \quad Q(f, g) = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \beta_j k(x_i, x'_j)$$

f et g sont bien dans $\mathcal{H}$, vu que ce sont des combinaisons linéaires d'éléments de $\mathcal{H}$.

Q 1.2.1 Montrez que Q peut s'exprimer uniquement à l'aide des β_j et $f(x'_j)$, ou des α_i et $g(x_i)$

Q 1.2.2 Montrez que $Q(f, g)$ est un produit scalaire de f et g (on pourra alors remplacer $Q(f, g)$ par $\langle f, g \rangle$). Pour cela, il s'agit de démontrer que :
- Q est symétrique
- Q est bilinéaire
- $Q(f, f) \geq 0$ (on montrera dans la dernière question que $Q(f, f) = 0 \iff f = 0$).

Q 1.2.3 Que vaut $Q(k(., x), f)$? $Q(k(., x), k(., x'))$? Justifiez le nom de k : *reproducing kernel.*

Q 1.2.4 En admettant que $Q(f, g)^2 \leq Q(f, f)^2 Q(g, g)^2$, montrez que $|f(x)|^2 \leq k(x, x).Q(f, f)$. Concluez.

Exercice 2 (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 2.1 On suppose un espace latent de dimension 4 : (comédie, romance, thriller, policier) et les données suivantes :

	comédie	romance	thriller	policier
films				
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
users				
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 2.1.1 Calculer la matrice de scores R pour ces données.

Q 2.1.2 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 2.1.3 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 2.2 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 2.3 Au terme trouvé à la question précédente, on ajoute le terme $\lambda(\sum_i ||u_{i,.}||^2 + \sum_j ||v_{.,j}||^2)$ pour établir la quantité finale à optimiser. Que représente les sommes de ce second terme ? Quel intérêt ?

Q 2.4 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 2.5 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 2.5.1 A quelle famille classique de problèmes d'apprentissage statistique correspond chacune des deux phases d'optimisation ?

Q 2.5.2 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 2.5.3 Donner l'implémentation python.

Q 2.6 Tous les utilisateurs n'ont pas la même moyenne d'avis (un utilisateur peut être plus sévère qu'un autre mais avoir les mêmes gouts), de même pour les items. Que manque-t-il au modèle pour pouvoir représenter ces biais ? Donner les modifications à apporter à l'algorithme.