Partiel ML/MLL - DAC

1 heure 30 - Barème indicatif - Document : 1 à 2 feuilles A4
Le barème est donné à titre indicatif pour indiquer l'importance relative des exercices.

Exercice 1 (7 points) – Ça hésite

Pour chaque question, une seule réponse est correcte. Donner vos réponses, **sans justifier**, sous la forme 1-V/1-F si l'affirmation 1 est vrai/fausse (0.25 points par bonne réponse, -0.25 points par fausse, barème indicatif).

1. La méthode des histogrammes est moins coûteuse en complexité pour l'inférence que les fenêtres de Parzen.
2. Les SVMs permettent de modéliser les probabilités a posteriori $P(y|x)$.
3. Il y a un gros risque de sur-apprentissage lorsqu'on utilise un classifieur optimal bayésien.
4. Limiter la taille d'un arbre de décision permet de limiter le sur-apprentissage.
5. Un k-nn avec $k = 3$ peut classifier parfaitement tout ensemble d'apprentissage linéairement séparable
6. L'erreur en test d'un k-nn avec $k = 1$ converge vers 0 quand le nombre d'exemples tends vers l'infini
7. Pour un jeu de données linéairement séparable, un SVM linéaire donne de meilleurs résultats sur l'ensemble d'apprentissage qu'un perceptron.
8. Pour un jeu de données linéairement séparable, un SVM linéaire donne de meilleurs résultats sur l'ensemble de test qu'un perceptron.
9. Le résultat d'une régression logistique dépend du point initial de la descente de gradient.
10. Plus l'ensemble d'apprentissage est grand, plus le risque de sur-apprentissage est petit.
11. La régression logistique apprend des frontières non linéaires car la fonction logistique est non linéaire.
12. La rétro-propagation permet d'apprendre un optimum global d'un réseau de neurones
13. La méthode des fenêtres de Parzen nécessite une discrétisation de l'espace.
14. La densité d'une variable aléatoire est toujours inférieure à 1.
15. Un arbre de décision peut séparer exactement tout ensemble de points disjoints
16. Une entropie strictement positive pour un ensemble de labels binaires indique la présence de plus de labels positifs que négatifs.
17. Selon l'algorithme de descente de gradient, il est toujours possible de trouver, si le gradient est non nul, un pas d'apprentissage permettant de faire décroître le coût à minimiser.
18. Lorsqu'on remarque qu'on sur-apprend en utilisant un K-NN, il vaut mieux augmenter K.
19. Il vaut mieux utiliser un K grand qu'un K faible pour un K-NN lorsqu'il y a beaucoup de bruit.
20. Certaines frontières apprises avec un perceptron ne peuvent être apprises par un SVM avec un noyau polynomial.
21. Il n'est pas possible de contrôler le sur-apprentissage dans la méthode des fenêtres de Parzen.
22. La vraisemblance d'un modèle fixé par rapport à des données baisse quand le nombre de données augmente.
23. Pour un jeu de données linéairement séparable, on obtient la même frontière de décision si on entraîne un SVM sur toutes les données ou seulement sur les vecteurs supports trouvés lors de l'apprentissage sur toutes les données.
24. Un coût $max(0, 1 - f(x)y)$ est généralement meilleur que $max(0, 0.1 - f(x)y)$ car il permet d'augmenter la marge des données avec les frontière de décision.
25. Pour apprendre un réseau de neurones, il suffit de connaître le gradient de la fonction de coût par rapport aux entrées de chaque couche et par rapport aux paramètres de chaque couche.
26. Lors d'une descente batch de gradient, il n'est pas nécessaire de mélanger les exemples.
27. Le gradient est toujours orthogonal à la tangente des isocontours d'une fonction.
28. Quand les données sont non linéaires, un réseau de neurones est toujours plus adapté qu'un SVM.

1-V, 3-F, 4-F, 5-V, 6-F, 7-v, 8-F, 9-V, 10-F, 11-F, 12-F,13-F,14-v,15-F,16-F,17-F,18-V,19-F,2O-V,21-V,22-V,23-F,24-F,25-V,26-V,27-F,28-V,29-V,30-F, 31-V, 32-F

Exercice 2 (5 points) – Ça Bayes

On considère un problème de classification binaire, où une observation x est engendré par l'une des deux lois suivantes : $p(x|y = 1) = \alpha_+ e^{-\alpha_+ x}$ et $p(x|y = -1) = \alpha_- e^{-\alpha_- x}$ pour $x \geq 0$, densités nulles pour $x \leq 0$ (avec α_- et α_+ strictement positif).

Q 2.1 (0.5 points) Vérifier que $p(x|y = 1)$ est bien une densité de probabilité.

Q 2.2 (1 point) On suppose disposer d'un jeu de données d'apprentissage $D = \{(x^i, y^i) \in (\mathbb{R}^+, \{-1, 1\})\}_{i=1}^N$. Donner l'expression de la vraisemblance de α_+ puis de la log-vraisemblance par rapport au jeu de données.

Q 2.3 (1 point) En déduire l'estimation de α_+ et α_- par maximum de vraisemblance.

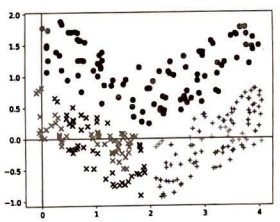
Q 2.4 (1 point) Quel est le classifieur optimal bayésien pour ce problème en considérant les classes équilibrées ? La frontière de décision ? On notera x_0 le point frontière de décision.

Q 2.5 (1.5 point) Calculer le risque 0-1 associé au classifieur optimal bayésien.

- $\int_0^\infty \alpha e^{-\alpha x} = [-e^{-\alpha x}]_0^\infty = 1$
- $L(D; \alpha_+) = \prod_{i=1|y^i=1}^N \alpha_+ e^{-\alpha_+ x^i} = (\alpha_+)^{N_+} e^{-\alpha_+ \sum_{i=1}^{N_+} x^i}$. Soit $LL(D; \alpha_+) = N_+ log(\alpha_+) - \alpha_+ \sum_{i|y^i=1} x^i$.
- Annulation de la dérivée par rapport à α_+ : $0 = N_+/\alpha_+ - \sum_{i|y^i=1} x^i$, donc $\alpha_+ = N_+ / \sum_{i|y^i=1} x^i$. Pareil pour α_-.
- $\frac{p(y=1|x)}{p(y=-1|x)} = \frac{p(x|y=1)}{p(x|y=-1)} = \frac{\alpha_+}{\alpha_-} e^{-(\alpha_+ - \alpha_-)x} = 1$, soit $x = -log(\alpha_-/\alpha_+)/(\alpha_+ - \alpha_-)$.
- On suppose $\alpha_+ > \alpha_-$ $\int_0^{x_0} p(y = +|x)dx + \int_{x_0}^\infty p(y = -|x)dx = 0.5([-e^{-\alpha_+ x}]_0^{x_0} + [-e^{-\alpha_- x}]_{x_0}^\infty) = 0.5(e^{-\alpha_- x_0} - e^{-\alpha_+ x_0}) = 0.5e^{x_0}(e^{-\alpha_-} + e^{-\alpha_+})$

Exercice 3 (5 points) – Ça perçoit

Q 3.1 On considère le problème de classification à 3 classes décrit dans la figure ci-jointe (deux dimensions x_1, x_2). On notera 1 la classe des ronds, 2 la classe des + et 3 la classe des ×.



Q 3.1.1 (1.5 points) Donner les 3 modèles linéaires qui permettent de séparer les classes 1 et 2, les classes 1 et 3, et les classes 2 et 3. Dessiner les séparatrices sur un graphique ainsi que les vecteurs des poids w_{12}, w_{13} et w_{23}. Les solutions sont-elles uniques ?

Q 3.1.2 (0.5 points) Est-il possible de construire un classifieur bayésien naïf qui permet de séparer les classes 1 et 2 sans toucher aux données ? Si oui, donner son expression.

Q 3.2 On considère maintenant que les classes 2 et 3 ne forment qu'une seule classe notée -1. L'objectif est donc de séparer la classe 1 de tous les autres points. Ceci ne peut être résolu par un modèle linéaire. On propose dans un premier temps de considérer une transformation des données.

Q 3.2.1 (0.5 points) Une première transformation envisagée est $\phi(\mathbf{x}) = ((x_1 + x_2)/2, (x_1 - x_2)/2)$. Sans calcul, expliquer en une phrase si elle est judicieuse.

Q 3.2.2 (1 point) Proposer une projection qui permet de rendre le problème linéairement séparable. Donner un modèle linéaire solution.

Q 3.3 (1.5 point) Proposer un réseau de neurone qui permet de résoudre le problème sans modification préalable des données.

Exercice 4 (4 points) – Ça tangue

A partir d'une description des conditions de navigation dans $\mathbb{R}^d$ on souhaite prédire y le cap de navigation d'un voilier exprimé en radian. On propose d'utiliser la fonction : $\mathcal{L}(y, \hat{y}) = 1 - cos(y - \hat{y})$ comme fonction de coût.

Q 4.1 (1 point) Pourquoi ne pas utiliser un coût aux moindres carrés ? Justifier que $\mathcal{L}$ est une fonction de coût adaptée.

Du fait de la périodicité, on peut pas utiliser la mse qui donnerait des écarts alors que les angles sont les mêmes. $\mathcal{L}$ vaut bien 0 si les angles sont les mêmes, et maximale (2) si les angles sont opposées.

Q 4.2 On propose d'utiliser un modèle linéaire f pour prédire $\hat{y}$ en fonction de $\mathbf{x} \in \mathbb{R}^d$.

Q 4.2.1 (0.5 point) Donner les paramètres de f et son expression.

$f_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i$

Q 4.2.2 (1.5 point) Donner un algorithme (complet) pour apprendre les paramètres de f à partir d'un jeu de données $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. (Rappel : $cos'(x) = -sin(x)$)

Descente de gradient sur $L = \sum_{i=1}^{n} 1 - cos(y^i - (w_0 + \sum_{j=1}^{d} w_j x_j^i))$,
$\frac{\partial L}{\partial w_j} = \sum_{i=1}^{n} x_j^i sin(y^i - (w_0 + \sum_{j=1}^{d} w_j x_j^i))$.

- Initialisation aléatoire de $\mathbf{w}$
- Pour *iter* de 1 à EPOCH : $\mathbf{w}^{iter} = \mathbf{w}^{iter-1} \epsilon * \nabla_{\mathbf{w}} L$

Q 4.3 (1 point) On souhaite modifier la fonction de coût afin de contrôler le sur-apprentissage. Que préconisez vous ? Quel(s) changement(s) apporter à l'algorithme d'apprentissage ?

pénalisation par $\frac{1}{2}\|w\|^2$, on ajoute λw au gradient précédent.

Examen Machine Learning - DAC

2 heures - Barème indicatif - Document : 1 à 2 feuilles A4
Le barème est donné à titre indicatif pour indiquer l'importance relative des exercices.

Exercice 1 (3 points) – Ni oui Ni non

Donner vos réponses, **sans justifier**, sous la forme 1-V par exemple si l'affirmation 1 est vrai, 1-F si l'affirmation est fausse. Chaque bonne réponse apporte 0.25 points, chaque mauvaise retire 0.25 points à votre note (ou moins, barème indicatif).

1. La somme de deux noyaux admissibles est un noyau admissible si et seulement si les projections sous-jacentes sont dans le même espace.
2. La somme de deux variables aléatoires gaussiennes suit une loi gaussienne seulement si elles ont la même espérance.
3. L'erreur en apprentissage diminue lorsque la VC-dimension de la famille de classifieurs considérée augmente.
4. Une matrice de covariance diagonale implique forcément l'indépendance des dimensions deux à deux.
5. Il existe des noyaux qui ne sont pas des produits scalaires de projections en dimension finie.
6. En RL, lorsque le MDP est parfaitement connu, il est possible de trouver la politique optimale sans exploration.
7. La valeur du discount factor γ n'a pas d'influence sur la politique optimale.
8. Dans le cas ou le MDP est connu, il vaut mieux estimer la Q-value que la V-value.
9. Dans le cas ou le MDP est inconnu, il vaut mieux estimer la Q-value que la V-value.
10. L'algorithme Value-iteration converge si le facteur de discount est dans $[0, 1[$
11. Les politiques trouvées par Policy-iteration sont meilleures que celles par Value-iteration
12. L'algorithme t-SNE est bien adapté pour construire des clusters significatifs.

1-F, 2-F, 3-V, 4-V, 5-V, 6-V, 7-F, 8-F, 9-V, 10-V, 11-F, 12-F

Exercice 2 (8 points) – Highway to gradient

L'architecture `Highway Network` a été proposé en 2015 spécifiquement pour les réseaux très profonds dédiés au traitement de l'image. Une couche de ce type de réseau est très semblable à celle d'un réseau classique mais réalise un mélange des entrées de la couche avec les sorties de cette couche.
Prenons par exemple une couche fully-connected non linéaire d'un réseau classique définie par la fonction $H(\mathbf{x}, \mathbf{W}_H) = \sigma(\mathbf{W}_H^t \mathbf{x} + b_H)$. Le Highway Network utilise une transformation $T(\mathbf{x}, \mathbf{W}_T) = \sigma(\mathbf{W}_T^t \mathbf{x} + b_T)$ afin de mélanger l'entrée $\mathbf{x}$ et la sortie usuelle de la couche $H(\mathbf{x}, \mathbf{W}_H)$: la sortie $\mathbf{y}$ de la couche est (avec $\odot$ l'opérateur de produit terme à terme)

$$\mathbf{y} = H(\mathbf{x}, \mathbf{W}_H) \odot T(\mathbf{x}, \mathbf{W}_T) + \mathbf{x} \odot (1 - T(\mathbf{x}, \mathbf{W}_T))$$

Q 2.1 (0.5 points) Supposons que l'entrée $\mathbf{x}$ soit de dimension d : $\mathbf{x} \in \mathbb{R}^d$. D'après la définition, quelles sont les dimensions de $\mathbf{W}_H$, $\mathbf{W}_T$, $\mathbf{y}$? (on supposera les biais b_H et b_T scalaires dans $\mathbb{R}$).

Même dimension que $\mathbf{x}$, d : $\mathbf{W}_H, \mathbf{W}_T \in \mathbb{R}^{d \times d}$, $\mathbf{y} \in \mathbb{R}^d$.

Q 2.2 (0.5 points) Calculez la dérivée de la fonction sigmoïde. Dans la suite, vous pouvez la noter $\sigma'(x)$ sans développer. Rappel : $\sigma(x) = \frac{1}{1+e^{-x}}$

$\sigma'(x) = \frac{e^{-x}}{(1+e^{-x})^2} = \sigma(x)(1 - \sigma(x))$

Q 2.3 On suppose un réseau constitué en premier d'une d'une couche Highway Network dont on notera la sortie $\mathbf{z}$, puis d'une couche linéaire M avec une fonction d'activation sigmoïde, et un coût aux moindres carrés :

- $\mathbf{z} = H(\mathbf{x}, \mathbf{W}_H) \odot T(\mathbf{x}, \mathbf{W}_T) + \mathbf{x} \odot (1 - T(\mathbf{x}, \mathbf{W}_T))$
- $\hat{\mathbf{y}} = \sigma(\mathbf{W}_M^t \mathbf{z} + b_M)$
- $L(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2}\|\mathbf{y} - \hat{\mathbf{y}}\|^2$

Q 2.3.1 (0.5 points) On suppose que $\mathbf{y} \in \mathbb{R}^p$. Quelles doivent être les dimensions de $\mathbf{W}_M$ et $\hat{\mathbf{y}}$? (on suppose que le biais b_M est scalaire dans $\mathbb{R}$).

$\mathbf{W}_M \in \mathbb{R}^{d \times p}$, $\mathbf{y} \in \mathbb{R}^p$

Q 2.3.2 (0.5 points) Calculez $\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{y}_i}$ la dérivée du coût par rapport à la i-ème sortie du réseau.

$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{y}_i} = -(y_i - \hat{y}_i)$

Q 2.3.3 (1 point) Calculez $\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{i,j}^M}$ pour un poids $w_{i,j}^M$ de $\mathbf{W}_M$.

$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{i,j}^M} = \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial w_{i,j}^M} = -(y_j - \hat{y}_j)\sigma'(\mathbf{W}_M^t \mathbf{z} + b_M)_j z_i$

Q 2.3.4 (1 point) Calculez $\delta_i = \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial z_i}$ pour la i-ème sortie $\mathbf{z}$ de la couche Highway.

$\delta_i = \sum_{j=1}^p \frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial \hat{y}_j} \frac{\partial \hat{y}_j}{\partial z_i} = -\sum_{j=1}^p (y_j - \hat{y}_j)\sigma'(\mathbf{W}_M^t \mathbf{z} + b_M)_j w_{i,j}^M$

Q 2.3.5 (1.5 point) Calculez pour un poids $w_{i,j}^T$ $\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{i,j}^T}$

$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{i,j}^T} = \delta_j \frac{\partial z_j}{\partial w_{i,j}^T} = \delta_j (H(x, \mathbf{W}_H)_j x_i \sigma'(\mathbf{W}_T \mathbf{x} + b_T)_j - x_j x_i \sigma'(\mathbf{W}_T \mathbf{x} + b_T))$

Q 2.3.6 (1.5 point) Calculez pour un poids $w_{i,j}^H$ $\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{i,j}^H}$

$\frac{\partial L(\mathbf{y}, \hat{\mathbf{y}})}{\partial w_{i,j}^T} = \delta_j \frac{\partial z_j}{\partial w_{i,j}^T} = \delta_j T(x, \mathbf{W}_T)_j x_i \sigma'(\mathbf{W}_H \mathbf{x} + b_H)_j$

Q 2.3.7 (1 point) Donnez l'algorithme d'optimisation du réseau.

Q 2.4 (bonus) À votre avis, quel(s) problème(s) permet de résoudre un réseau Highway et pourquoi ?

Exercice 3 (6 points) – Couplage de SVMs

Soit deux problèmes de classification, le premier consistant à discriminer les images de chiffres entre des 1 et des 3, le deuxième entre des 3 et de 8. On dispose pour cela de deux ensembles d'apprentissage : $\mathcal{D}_1 = \{(\mathbf{x}_{i1}, y_{i1}), i1 = 1, \ldots, n\}$ composé de 1 et de 3 et $\mathcal{D}_2 = \{(\mathbf{x}_{i2}, y_{i2}), i1 = 1, \ldots, n\}$ composé de 3 et de 8. On souhaite apprendre deux classifieurs linéaires, le premier $f_1(\mathbf{x}) = \mathbf{w}_1 \cdot \mathbf{x}$ en charge de séparer les 1 des 3, le deuxième $f_2(\mathbf{x}) = \mathbf{w}_2 \cdot \mathbf{x}$ en charge de séparer les 3 des 8. Vu la similarité des problèmes, on sup pose que les solutions doivent être également similaires. Pour représenter cette similarité, on définit un paramètre $\mathbf{w}_0$ commun aux deux problèmes et les deux paramètres $\mathbf{w}_1$ et $\mathbf{w}_2$ comme des variations autour de ce paramètre :

$$\mathbf{w}_1 = \mathbf{w}_0 + \mathbf{v}_1 \quad \text{et} \quad \mathbf{w}_2 = \mathbf{w}_0 + \mathbf{v}_2$$

En reprenant la formulation du SVM, la fonction à minimiser sous certaines contraintes est :

$$min_{\mathbf{w}_0,\mathbf{v}_1,\mathbf{v}_2,\xi_1,\xi_2} \frac{C_0}{2}(\|\mathbf{v}_1\|^2 + \|\mathbf{v}_2\|^2) + \|\mathbf{w}_0\|^2 + C\sum_{i=1}^{n}(\xi_{i1} + \xi_{i2})$$

Q 3.1 (0.5 point) A quoi servent les ξ_{ij} ? Quelles contraintes doit on poser pour compléter le problème de minimisation ?

$y_{ij}(\mathbf{w}_0 + \mathbf{v}_j)\mathbf{x}_{ij} \geq 1 - \xi_{ij}$ pour $i = 1, \ldots, n$ et $j = 1, 2$
$\xi_{ij} \geq 0$ pour $i = 1, \ldots, n$, $j = 1, 2$

Q 3.2 (0.5 point) Que représente C et C_0 ? Comment les choisir ?

C coefficient pour l'importance des erreurs, C_0 coefficient pour pénaliser l'écart à $\mathbf{w}_0$. Plus C_0 petit, plus les svms seront communs .

Q 3.3 (0.5 point) Montrer que le Lagrangien peut s'écrire de la manière suivante et donner les contraintes sur α_{ij}, γ_{ij}

$$\mathcal{L} = \frac{1}{2}\|\mathbf{w}_0\|^2 + \frac{C_0}{2}\sum_{j=1}^{2}\|\mathbf{v}_j\|^2 + C\sum_{j=1}^{2}\sum_{i=1}^{n}\xi_{ij} - \sum_{j=1}^{2}\sum_{i=1}^{n}\alpha_{ij}[y_{ij}(\mathbf{w}_0 + \mathbf{v}_j)\cdot\mathbf{x}_{ij} - 1 + \xi_{ij}] - \sum_{j=1}^{2}\sum_{i=1}^{n}\gamma_{ij}\xi_{ij}$$

On ajoute un terme par contrainte de $\xi_{ij} \geq 0$ avec coef lagrangien $\gamma_{ij} \geq 0$ (donc $2n$ termes) et un terme par contrainte sur $y_{ij} < (w_0 + v_j > x_{ij} \geq 1 - \xi_{ij}$ avec coef $\alpha_{ij} \geq 0$ ($2n$ termes egalement).

Q 3.4 (1.5 point) Quelles sont les conditions d'optimalité par rapport aux variables primales $\mathbf{w}_0$, $\mathbf{v}_j$ et ξ_{ij} ?

$\nabla_{\mathbf{w}_0}\mathcal{L} = 0 = C_0\mathbf{w}_0 - \sum_{ij}\alpha_{ij}(y_{ij}\mathbf{x}_{ij})$, donc $\mathbf{w}_0 = \frac{1}{C_0}\sum_{ij}\alpha_{ij}(y_{ij}x_{ij})$
$\nabla_{\mathbf{v}_j}\mathcal{L} = 0 = \mathbf{v}_j - \sum_{i=1}^{n}\alpha_{ij}y_{ij}x_{ij}$ donc $\mathbf{v}_j = \sum_{i=1}^{n}\alpha_{ij}y_{ij}x_{ij}$
$\nabla_{\xi_{ij}}\mathcal{L} = 0 = C - \alpha_{ij} - \gamma_{ij}$ donc $\alpha_{ij} + \gamma_{ij} = C$

Q 3.5 (0.5point) Quelles sont les expressions de $\mathbf{w}_0$ et $\mathbf{v}_j$?

cf ci-dessus

Q 3.6 (0.5 point) Donner un encadrement des α_{ij}.

du coup $0 \leq \alpha_{ij} \leq C$ comme $\gamma_{ij} \geq 0$

Q 3.7 (1 point) Quel est le problème dual correspondant, en particulier l'expression de $\mathcal{L}$ sans $\mathbf{w}_0$, $\mathbf{v}_j$?

$\mathcal{L} = \frac{C_0}{2}\frac{1}{C_0^2}\sum_{ij,i'j'}\alpha_{ij}\alpha_{i'j'}(y_{ij}y_{i'j'}x_{ij}x_{i'j'}) + \frac{1}{2}\sum_{j=1}^{2}\sum_{i=1}^{n}\sum_{i'=1}^{n}\alpha_{ij}\alpha_{i'j}y_{ij}y_{i'j}x_{ij}x_{i'j}$

Q 3.8 (1 point) Exprimer $f_1(\mathbf{x})$ et $f_2(\mathbf{x})$ en fonction des α_{ij}.

Exercice 4 (5 points) – Déphasé

On considère le modèle suivant $y = f(x) + \epsilon$ avec $f(x) = \sum_{j=1}^{d} w_j sin(cx + \phi_j)$, avec $\mathbf{w} \in \mathbb{R}^d$, $\phi \in \mathbb{R}^d$, $c, x \in \mathbb{R}$ et on considère que ϵ suit une loi normale centrée en 0 et de variance $1/\alpha^2$.

Q 4.1 (0.5points) Donner un argument pour montrer que y suit une loi normale quand on fixe x, c, $\mathbf{w}$, ϕ et α. Donner l'expression de $p(y|x, c, \alpha, \mathbf{w}, \phi)$.

$f(\mathbf{x})$ est déterministe, donc à x fixé, y est la somme d'une constante et d'une loi normale, donc $p(y|f(\mathbf{x})) = p(y|x, c, \mathbf{w}, \alpha, \phi) = N(f(x), 1/\alpha^2)$.

Q 4.2 (1.5 points) Soit un ensemble de données $\mathcal{D} = \{x^i, y^i\}_{i=1}^n$, quelle est la vraisemblance du modèle par rapport à $\mathcal{D}$? Donner également l'expression de la log-vraisemblance.

$L = \prod_{i=1}^{n} p(y|x, c, \mathbf{w}, \alpha) = \prod_{i=1}^{n} \frac{1}{K} e^{-\|f(x)-y\|^2\alpha^2}$, soit en passant au log, $NLL = -\sum_{i=1}^{n}\|f(x) - y\|^2 - n\alpha^2 = -\sum_{i=1}^{n}\sum_{j=1}^{d}(y^i - w_j sin(cx^i + \phi_j))^2 - n\alpha^2$

Q 4.3 (2 points) Donner un algorithme pour optimiser $\mathbf{w}$ lorsque tous les autres paramètres sont connus.

$\frac{\partial NLL}{\partial w_j} = -2\sum_{i=1}^{n}(y^i - w_j sin(cx^i + \phi_j))sin(cx^i + \phi_j) = 0$, donc $w_j(\sum_{i=1}^{n} sin(cx^i + \phi_j)^2 = \sum_{i=1}^{n} y^i sin(cx^i + \phi_j)$, donc $w_j = \frac{\sum_{i=1}^{n} y^i sin(cx^i + \phi_j)}{\sum_{k=1}^{n} sin(cx^k + \phi_j)^2}$

Q 4.4 (1 point + bonus ce qu'on veut) (bonus) Et si on voulait également apprendre les ϕ, quelle approche préconiseriez vous ? Discuter des problèmes potentiels.

Aucune idée. C'est pour ceux qui ne savent plus quoi faire.