le cnam

Sujet UE RCP209
Apprentissage, réseaux de neurones et modèles graphiques

Année universitaire 2018–2019

Examen de rattrapage : 12 avril 2019

Responsable : Michel CRUCIANU

Durée : 2h30

Seuls documents autorisés : 2 pages A4 recto-verso, manuscrites.

Les téléphones mobiles et autres équipements communicants (PC, tablette, etc.) doivent être éteints et rangés dans les sacs pendant toute la durée de l'épreuve. Calculatrice autorisée.

Sujet de 5 pages, celle-ci comprise.

Vérifiez que vous disposez bien de la totalité des pages du sujet en début d'épreuve et signalez tout problème de reprographie le cas échéant.

---

1. Expliquez le principe de la recherche en grille (*grid search*) avec validation croisée pour le choix des valeurs des hyperparamètres. (**2 points**)

   Correction : Les hyperparamètres (par ex. constantes de régularisation, types de régularisation employés, etc.) permettent de contrôler la procédure de modélisation. Pour choisir les valeurs des hyperparamètres, il est nécessaire de comparer un ensemble de modèles obtenus chacun avec des valeurs différentes pour les hyperparamètres et de conserver le modèle qui présente l'erreur de validation croisée la plus faible (donc la meilleure généralisation espérée). Pour chaque hyperparamètre, on définit un ensemble de valeurs à explorer ; avec n hyperparamètres, on obtient ainsi une « grille » k-dimensionnelle dans laquelle chaque point représente une combinaison de valeurs pour les différents hyperparamètres. Pour chaque point de la grille on développe k modèles par validation croisée *k-fold*. Le point de la grille pour lequel l'erreur de validation croisée est la plus faible indique les meilleures valeurs des n hyperparamètres.

2. Arbres de décision (**2 points**)
   - Décrivez brièvement le principe de construction des arbres de décision par l'algorithme ID3.
   - Comment gèrent les arbre de décision le problème des données manquantes ?

   Correction :
   - L'algorithme commence par le placement de tous les exemples d'apprentissage dans le nœud racine. Ensuite, chaque nœud est coupé sur un des attributs restants (qui n'a pas encore été testé). Le choix de cet attribut se fait à travers une mesure d'homogénéité par rapport à la variable cible. Cette mesure est le gain d'information obtenu par le découpage. Le processus s'arrête quand les éléments d'un nœud ont la même valeur pour la variable cible (homogénéité).
   - Dans le cas de données manquantes, la technique la plus utilisée est celle des *surrogate splits* ou variables-substituts : l'opération continue sur un autre attribut qui, à l'apprentissage, a donné un découpage (split) similaire.

3. Forets aléatoires (**2 points**) Comment on évalue l'importance des attributs dans les algorithmes de type *bagging* ou forêts aléatoires ?

   Correction : Les attributs peuvent être évalués pour voir leur contribution dans la construction de l'arbre (mesure de Gini) ou pour tester leur robustesse aux erreurs de capteurs ou le bruit sur la classification (erreur OOB) :
   - Gini : Le changement dans l'impureté (ou gain d'information) dans chaque noed cumulé sur tous les arbres de la forêt.

— Erreur OOB : tous les échantillons OOB sont évalués par l'arbre et l'erreur mesurée. Ensuite on permute aléatoirement les valeurs sur chaque attribut j et on mesure le taux d'erreur à nouveau. La valeur finale est la dégradation moyenne (changement du taux d'erreurs) sur tous les arbres.

4. Machines à vecteurs de support (SVM) (**2 points**)
   — Expliquez brièvement comment les SVM gèrent le cas ou les données sont non séparables linéairement.
   — Pourquoi on préfère résoudre le problème dual plutôt que le primal ?

   Correction :
   — Pour gérer ce type de problème on utilise une technique dite de *marge souple*, qui tolère les mauvais classements et qui consiste en rajouter des variables de relâchement des contraintes ξ_i (qui mesurent l'écart des erreurs via une fonction de perte) et pénaliser ces relâchements dans la fonction objectif.
   — Le problème dual est un problème d'optimisation quadratique, pour lequel il existe des algorithmes très performants. Aussi, la formulation duale fait apparaître la matrice de Gram, ce qui permet de gérer le cas non linéaire à travers des algorithmes à noyaux.

5. Algorithmes à noyau (**2 points**)
   — Soit $x, y \in \mathbb{R}$, le noyau $K(x, y) = e^{e^{x+y}}$ est un noyau défini positif ?
   — Pour quel type d'algorithmes peut-on construire une version à noyau en utilisant le *kernel trick* ?

   Correction :
   — Oui, car e^{x+y} est un noyau conforme et l'exponentielle d'un noyau défini positif est un noyau défini positif.
   — Tout algorithme qui utilise seulement des produits scalaires entres les échantillons de données peut tout de suite être appliqué dans l'espace de Hilbert H, car le produit scalaire dans cet espace se calcule directement via le noyau $K(x, y) = \langle \phi(x), \phi(y) \rangle$, sans avoir besoin d'expliciter la projection $\phi(\cdot)$.

6. Expressivité (**3 points**)
   — Que signifie la propriété « d'approximation universelle » des réseaux de neurones d'un point de vue de l'expressivité du modèle ?
   — Le pouvoir expressif d'un réseau à 10 couches cachées est-il supérieur à celui d'un réseau à une couche cachée ?
   — Quel est alors l'intérêt de construire un réseau profond ?

Correction :

— On peut approximer n'importe quelle fonction continue entre l'entrée et la sortie avec une précision arbitraire.

— Non, un réseau à une couche caché est le même que celui à 10 couche cachés.

— Le réseau profond va avoir un pouvoir expressif équivalent avec moins de paramètres, et va donc potentiellement mieux généraliser.

7. Optimisation (**3 points**)

— En quoi consiste une méthode de descente de gradient stochastique ? Quel est son intérêt pour l'entraînement des réseaux de neurones ?

— En quoi le mauvais conditionnement de la matrice Hessienne de la fonction objectif est-il problématique pour les méthodes de descente de gradient ? Préciser la solution apportée à ce problème par la méthode de « momentum ».

Correction :

(a) Approximer le gradient sur un batch pour avoir des mises à jour plus fréquentes qu'avec le dataset complet ; crucial dans le cas de réseaux profonds.

(b) Le mauvais conditionnement induit des oscillations très lentes dans la direction « intéressante » et très rapides dans la directions « inintéressantes ». Le momentum par son effet de mémoire temporelle va annuler des oscillations contradictoires dans le directions inintéressantes et renforcer les petites variations consistantes dans les directions intéressantes.

8. Transfert (**2 points**)

— Expliquer en quoi consistent les « Deep Features » et préciser leur importance.

— Détailler comment les « Deep Features » peuvent être utilisées une tâche de localisation d'objet, *e.g.* méthode R-CNN.

Correction :

— DF : réseaux pré-entraînés sur ImageNet. Très important car permet d'aborder des tâches avec données peu massivement annotées.

— Region proposal + DF + multi task (classif + BB regression).

9. Auto-différentiation (**1 point**)

— Quel est le principe de « l'auto-différentiation », et sa différence par rapport aux méthodes de différentiation symbolique et numérique (4 lignes max) ?

Correction : C'est un intermédiaire entre les méthodes de différentiation symbolique et différentiation numérique : on fait de la différentiation symbolique mais au niveau d'opérateurs élémentaires, puis on calcule le résultats numérique des gradients que l'on propage ensuite.

10. Prédiction structurée (**1 point**). Quelle est la différence entre les méthodes d'apprentissage structuré et les méthodes de classification ? (4 lignes max).

    Correction : La prédiction structurée permet d'entraîner des modèles à prédire des sorties discrètes quelconques, beaucoup plus générales que les méthodes de classification (limitées à prédire une classe en sortie). En particulier, l'intérêt est de pouvoir modéliser la corrélation entre les variables de sortie.