IA et science des données

Cours 11 – mardi 23 avril 2024
Retour au supervisé. Corrections.

Christophe Marsala

Sorbonne Université

LU3IN026 - 2023-2024

Plan du cours

Retour au supervisé : méthodes d'ensembles

Exercices

Biais et Variance (1)

- Apprentissage : trouver f, fonction de prédiction, telle que :

$$y = f(\mathbf{x}) + \epsilon$$

avec $\epsilon \geq 0$ le plus petit possible
- idéalement : $\epsilon = 0$ (mais on n'y arrive jamais...)
- la "forme" de f est importante : elle utilise les variables de $\mathbf{x}$
  - linéaire, quadratique,...
  - arbre de décision
  - ...
- Modèle parcimonieux : nombre réduit de variables utilisées,...
  - idée : modèle parcimonieux $\Longrightarrow$ faible variance
- Biais : complexité du modèle
- Variance : capacité du modèle à changer si la base d'apprentissage change

Biais et Variance (2)

- Objectif : faible biais & variance faible
  - très difficile d'atteindre les 2... il faut choisir !
- Nouvelle approche : réduire la variance
  - combiner plusieurs classifieurs
  - agréger leur résultats pour améliorer les performances
- Différentes façons de faire
  - on regarde avec les arbres (par exemple)
  - multiplier les arbres pour les combiner ensuite

L'approche BAGGING

- Bootstrap AGGregatING
- Construire un ensemble de classifieurs de même type
- Agréger leurs résultats lors d'une classification
- $\Longrightarrow$ approche très efficace !
  - la variance globale est plus faible que la variance de chaque classifieur
- Si les classifieurs sont des arbres de décision : forêt

L'approche BAGGING : apprentissage et classification

Apprentissage :

- Soit $\mathbf{X}$ une base d'apprentissage avec n exemples
- Soit B le nombre de classifieurs souhaités et $m < n$ le nombre d'exemples à choisir
  1. Extraire B sous-bases de $\mathbf{X}$: $\mathbf{X}_1, \ldots, \mathbf{X}_B$
     - sélection aléatoire de m exemples de $\mathbf{X}$
     - avec ou sans remise
  2. Construire un classifieur f_k pour chaque sous-base $\mathbf{X}_k$
- Au final : on obtient un ensemble de B classifieurs $f_1, \ldots, f_B$

Classification :

- Soit un ensemble de B classifieurs $f_1, \ldots, f_B$
- Soit un exemple $\mathbf{x}$ à classer
  1. calculer $f_k(\mathbf{x})$ pour chaque classifieur $k = 1, \ldots, B$
  2. classe finale prédite de $\mathbf{x}$: classe majoritaire parmi les $f_k(\mathbf{x})$

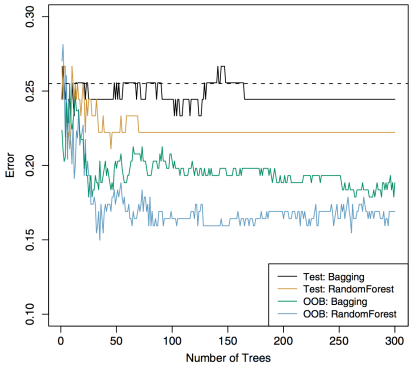
Les forêts aléatoires (random forest)

- Idée : plus les arbres sont diversifiés, meilleur sera le score global
- Augmenter la diversité : choisir aléatoirement les variables à utiliser !
- Bagging modifié : random forest
- Soit $\mathbf{X}$ une base d'apprentissage avec n exemples
- Soit B le nombre de classifieurs souhaités, $m < n$ le nombre d'exemples à choisir et $p \leq d$ variables de description à choisir
  1. Extraire B sous-bases de $\mathbf{X}$: $\mathbf{X}_1, \ldots, \mathbf{X}_B$
     - sélection aléatoire de m exemples de $\mathbf{X}$
     - sélection aléatoire de p variables de descriptions
  2. Construire un classifieur f_k pour chaque sous-base $\mathbf{X}_k$
- Remarque : B, m et p sont des hyper-paramètres de l'algorithme

Evaluation d'ensembles

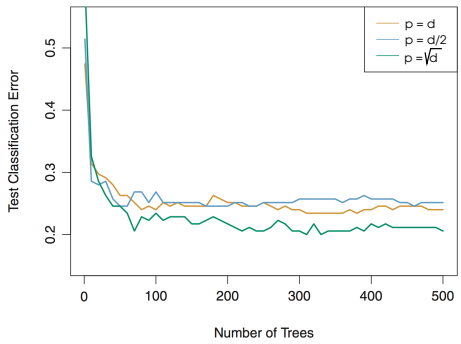
- Pour évaluer un ensemble construit par Bagging / random forest
- Validation croisée
  - très coûteuse pour évaluer un ensemble
  - il faut construire B classifieurs à chaque fois !
- Evaluation Out Of the Bag (OOB)
  - adaptée aux ensembles et suffisante pour les évaluer
  - évaluer f_k sur les exemples de $\mathbf{X}$ non sélectionnés pour le construire
  - chaque $\mathbf{x}$ est évalué par les f_k pour lesquels il n'a pas été utilisé en apprentissage
  - compter le nombre de fois où il est bien classé sur le nombre de fois où il est classé

Performances bagging vs random forest



(source : "An introduction to statistical learning", Gareth et al.)

Performances random forest : choix du nombre d'attributs



(source : "An introduction to statistical learning", Gareth et al.)

Ensembles de classifieurs

- Approches de construction d'ensembles
  - bagging : bootstrap aggregating
  - random forests
- Evaluation :
  - validation croisée
  - approche Out Of Bag

Plan du cours

Retour au supervisé : méthodes d'ensembles

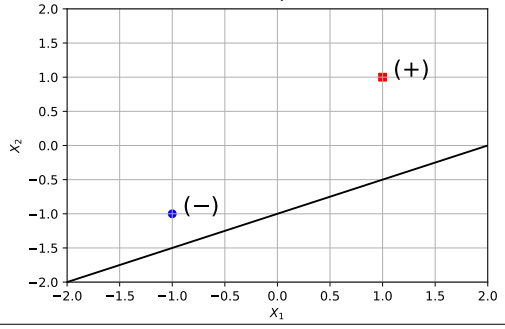
Exercices

Exercice 1 : Apprentissage supervisé, frontière

1. Soit $\mathbf{w} = (w_1, \ldots, w_d)$ un vecteur de poids à valeurs dans $\mathbb{R}$ utilisé pour prendre une décision linéaire. Donner l'expression permettant de calculer le produit scalaire de chaque exemple de $\mathbf{X}$ avec $\mathbf{w}$.
2. On se place maintenant en langage Python, on note X le numpy.array qui contient $\mathbf{X}$ et w le numpy.array qui contient $\mathbf{w}$. Donner les instructions python pour calculer le produit scalaire de tout vecteur de $\mathbf{X}$ par $\mathbf{w}$, sans utiliser de boucle.

Exercice 1 : Apprentissage supervisé, frontière (suite)

3. Soit un jeu de données supervisé $(\mathbf{X}, Y)$. Les données et la frontière de décision associée à $f(\mathbf{x}) = \mathbf{w}\mathbf{x} + b$ sont représentées ci-contre. Donner les valeurs de d et n correspondantes, puis donner les valeurs de $\mathbf{w}$ et b associés au tracé de la frontière (plusieurs valeurs sont possibles, ...).



Exercice 2 : Clustering

1. Montrer que la distance de Manhattan est bien une mesure de distance.
2. Dans le cours, des approches ont été données pour calculer la distance entre 2 clusters ...
3. En utilisant la distance euclidienne et l'approche par centre de gravité, appliquer à la main l'algorithme de clustering hiérarchique, méthode par agglomération, sur les données fournies sur le transparent 21 ...
4. On considère la base d'apprentissage de $[0, 10] \times [0, 10]$ contenant les 7 exemples suivants : $\mathbf{X} = \{(1, 2), (1, 4), (3, 4), (3, 5), (6, 2), (6, 5), (8, 3)\}$ (remarque : on considère que cette base est déjà normalisée) ...

Examen 2022 : Évaluation black box

Soit un problème de classification binaire équilibré réputé difficile reprenant encore une fois les notations précédentes. Un expert métier vous indique qu'une performance de 90% est souhaitée mais que les résultats plafonne actuellement à 60%. Un des ingénieurs de votre équipe a eu accès à un modèle très performant (`blackbox_xgb`) et vous propose le code (fonctionnel) suivant :

```
// chargement des donnees X, Y etc...
mod  = blackbox_xgb.train(X,Y)
pred = blackbox_xgb.predict(X)
perf = np.where(pred == Y, 1, 0).mean()

if perf >= 0.9:
  print("Bravo, la solution est prete pour la commercialisation")
elif perf >= 0.6:
  print("OK, vous avez atteint la performance de base")
else :
  print("Echec: votre approche n'atteint pas les performances de reference")
```

Q. 1. Que pensez-vous du travail de votre ingénieur ? Pourquoi ?

Q. 2. (bonus culturel) Quelle est cette approche mystère selon vous ?