Données multimédia et spatio-temporelles (NFE205)

AutoML : automatisation de l'apprentissage statistique

Michel Crucianu
(prenom.nom@cnam.fr)

Département Informatique
Conservatoire National des Arts & Métiers, Paris, France

1 octobre 2021

Pourquoi AutoML ?

- Le développement de modèles prédictifs par apprentissage statistique à partir de données demande des ressources et des connaissances :
  - Données : hétérogénéité, représentation inadaptée, valeurs manquantes
  - Familles de modèles : grande diversité, exigences spécifiques concernant la représentation des données, différences de fonctionnement, de coût
  - Apprentissage : nombreux hyper-paramètres dont il faut choisir les meilleures valeurs
  - Apprentissage : plusieurs techniques de combinaison de modèles à explorer pour améliorer les résultats
  - Déploiement : encapsulation de la chaîne de traitement complète pour mise en production

→ Automatiser le plus possible le développement de modèles permettrait

- Aux *data scientists* de se concentrer sur l'analyse du problème à modéliser et de son évolution, en dialogue avec les experts métier
- Aux organisations d'élargir l'application de modèles prédictifs et d'aller plus vite de l'analyse au déploiement

Plan du cours

Construction et mise en œuvre d'un modèle décisionnel : étapes

(en rouge ce que AutoML prend en charge dans une certaine mesure)

1. Collecte et préparation des données
   - Conception du recueil de données, recueil de données
   - Nettoyage, vérification et encodage des données
   - Imputation des données manquantes
2. Choix des objectifs et des métriques à optimiser
3. Sélection et transformation de variables (*feature selection and engineering*)
4. Éventuel choix amont des techniques de modélisation à employer (arbres de décision, SVM, forêts aléatoires, réseaux de neurones, etc.)
5. Si réseaux de neurones profonds employés, définition des architectures à explorer

Construction et mise en œuvre d'un modèle décisionnel : étapes (2)

6. Construction de plusieurs modèles
   - Pour modèles individuels, optimisation des paramètres sur données d'apprentissage
   - Sélection des hyper-paramètres sur données de validation (possible délimitation manuelle des données de validation)
   - Combinaison de modèles par *bagging* et/ou *boosting*
   - Combinaison de modèles par *stacking*
7. Éléments d'explication : importance des variables (globale ou locale), visualisations
8. Estimation des performances futures attendues des modèles développés
   - Possible délimitation manuelle des données de test, par ex. si non stationnarité marquée
9. Déploiement
   - Extraction des variables explicatives retenues par les modèles
   - Encapsulation de la chaîne de traitement (*pipeline*) pour mise en production
10. Post-déploiement
    - Suivi des performances (*monitoring*) : exige des évaluations périodiques sur de nouvelles données étiquetées
    - Mise à jour des modèles (*model update*)

Plan du cours

Collecte et préparation des données

- Nettoyage, vérification :
  - Détection et adaptation du codage (ISO, unicode, etc.)
  - Suppression de valeurs anormales (par ex. caractères non numériques pour variables numériques, valeurs aberrantes)
- Encodage des données :
  - Détection du type des variables (quantitatives, nominales), représentation adaptée au type et aux exigences des techniques de modélisation
  - Encodage (*embedding*) des textes (FastText, BERT, GPT) et des images (ResNet xx)
- Imputation des données manquantes :
  - L'absence de certaines valeurs pour des variables explicatives est fréquente
  - Suppressions lignes (observations) à valeurs manquantes $\Rightarrow$ réduction du nombre de données, introduction de biais de modélisation

  $\rightarrow$ Imputation : estimation des valeurs manquantes à partir des valeurs présentes des mêmes variables et/ou des autres variables explicatives, utilisation de ces valeurs estimées pour la suite de la modélisation

Méthodes de combinaison de modèles

- Principe : apprentissage de plusieurs modèles de base (*weak learners*, le plus souvent de même type) combinés ensuite pour obtenir le modèle final

1. *Bagging* :
   - Chaque modèle de base est appris indépendamment, sur un échantillon des observations (et éventuellement un échantillon des variables)
   - Combinaison (pondérée) : vote majoritaire si classification, moyenne si régression
2. *Boosting* :
   - Modèles de base (en général de même type) appris en séquence, sur les mêmes variables et observations, chaque modèle cherche à « corriger » les erreurs des précédents
   - Combinaison (pondérée) : vote majoritaire si classification, moyenne si régression
3. *Stacking* :
   - Plusieurs modèles de base, en général hétérogènes, sont appris indépendamment, ensuite un méta-modèle apprend à prédire à partir de leurs prédictions
   - Combinaison à travers le méta-modèle appris

Optimisation des valeurs des hyper-paramètres

- Hyper-paramètres :
  - Structure : architecture réseau de neurones, nombre d'arbres dans forêt aléatoire, etc.
  - Régularisation : pondération L_2 (ou/et L_1), taux de *drop-out*, etc.
- Difficulté de l'optimisation :
  - Dépendance complexe entre performance(s) et valeurs des hyper-paramètres
  - Nombre parfois élevé d'hyper-paramètres ($\rightarrow$ malédiction de la dimension)
- Approches d'optimisation (voir partie 1 de `https://www.automl.org/book/`) :
  1. Recherche en grille (*grid search*) hiérarchique : optima pas nécessairement sur la grille (même hiérarchique), malédiction de la dimension
  2. Recherche aléatoire : meilleure efficacité mais sous-optimale
  3. Recherche adaptative : modélisation progressive des régions aux performances élevées ; peut être initialisée par des résultats déjà obtenus sur des *datasets* similaires

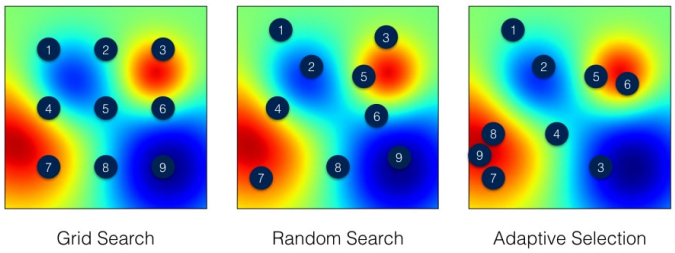

FIG. – Trois approches dans l'optimisation des hyper-paramètres (source de l'image : blog.ml.cmu)

Plan du cours

Offre AutoML : typologie

1. Outils historiques d'origine académique :
   - *Open source*, méthodes état de l'art mais visualisation peu présente et scalabilité limitée
   - Exemples : AutoKeras, Auto-sklearn, AutoWEKA, TPOT
2. Outils issus de *start-ups* qui commercialisent des services associés :
   - Mettent en avant des outils de visualisation et d'assistance
   - Ne sont pas *open source* mais font appel à des composants externes *open source*
   - Disponibles pour la plupart sur *cloud*, scalabilité toutefois variable
   - Exemples : DataRobot, dotData, H2O AutoML, Tazi.ai
3. Outils issus de grandes entreprises non spécialisées en IA :
   - Développés pour usage interne, ensuite publiés en *open source*
   - Exemples : Uber Ludwig, Salesforce TransmogrifAI
4. Outils issus de fournisseurs de services de *cloud* :
   - Mettent en avant principalement la scalabilité
   - La plupart ne sont pas *open source*
   - Exemples : Amazon Autopilot, AWS AutoGluon, MS Azure AutoML, Google AutoML

- Évaluations comparatives (partielles) : [2], [1], `https ://www.automl.org/book/`

Offre AutoML : une comparaison (incomplète)

Nom	Open source	Fonctionnalités[1]	Images	Scalabilité
Autopilot (AWS)	-	7	-	cloud
AutoGluon (AWS)	oui	8	oui	GPU
AutoKeras	oui	7	oui	GPU
Auto-sklearn	oui	5	-	-
Azure AutoML	-	8	oui	cloud
Dataiku	-	7	-	cloud
DataRobot	-	8	-	cloud
dotData	-	8	-	cloud
Google AutoML	-	9	oui	cloud
H2O AutoML	oui	7	-	cloud
Ludwig (Uber)	oui	8	oui	GPU
Tazi.ai	-	7	-	-
TPOT	oui	5	-	Dask
TransmogrifAI	oui	6	-	Spark

[1] Une partie de l'évaluation est issue de Targetbase.
Tous traitent des données tabulaires, ainsi que (potentiellement) textuelles à travers des *embeddings*.

Plan du cours

AutoGluon : particularités

(`https ://auto.gluon.ai/stable/index.html` par Amazon Web Services, en *open source*)

- Qu'est-ce qui est automatisé :
  - Encodage données, imputation valeurs manquantes
  - Construction de plusieurs types de modèles, exploration d'architectures de réseaux de neurones, réglage hyper-paramètres, combinaison modèles (*bagging/boosting/stacking*)
  - Évaluation de l'importance des variables
  - Déploiement et mise à jour des modèles
- Types de données : données tabulaires (variables continues et nominales), textes, images et combinaison des trois (données multimodales)
- Types de tâches : apprentissage supervisé (classification, régression, détection d'objets), apprentissage par renforcement
- Déploiement : distillation possible pour réduire le coût du modèle en production
- Interface basique : *notebook* Python
- Prise en main relativement rapide sans expertise en apprentissage

AutoGluon : utilisation

- Étapes dans la construction d'un modèle :
  1. Lecture des données à partir de fichier(s) (avec `pandas`)
     → AutoGluon lit les données, identifie le type de chaque variable et adapte son codage
  2. Séparation d'un ensemble de données de test et lancement de l'apprentissage
     → AutoGluon détermine le type de problème (classification ou régression), fait l'imputation des données manquantes, développe des modèles de différents types et ensuite des combinaisons de modèles ; la séparation apprentissage | validation est faite par AutoGluon ou l'utilisateur
  3. Évaluation des modèles obtenus sur données de test, avec plusieurs métriques adaptées
     → AutoGluon évalue les modèles et combinaisons de modèles pour retourner leurs performances
  4. Évaluation de l'importance des différentes variables explicatives pour le modèle choisi
     → AutoGluon évalue chacune des variables explicatives (en permutant aléatoirement ses valeurs) et retourne un classement des variables qui aide à mieux comprendre les prédictions du modèle
- Les modèles de base considérés par AutoGluon pour données tabulaires :
  `KNeighbors`, `LightGBM`, `RandomForest`, `CatBoost`, `ExtraTrees`, `XGBoost`, `NeuralNetFastAI`, `NeuralNetMXNet`, `WeightedEnsemble_L2`

→ Voir les détails dans les travaux pratiques avec AutoGluon

Références I

N. Erickson, J. Mueller, A. Shirkov, H. Zhang, P. Larroy, M. Li, and A. Smola.

AutoGluon-tabular : Robust and accurate AutoML for structured data.
arXiv preprint arXiv :2003.06505, 2020.

A. Truong, A. Walters, J. Goodsitt, K. E. Hines, C. B. Bruss, and R. Farivar.

Towards automated machine learning : Evaluation and comparison of automl approaches and tools.
CoRR, abs/1908.05557, 2019.