Introduction au Machine Learning

C4 - Méthodologie

Lucas Gnecco Heredia

LAMSADE - Université Paris-Dauphine
lucas.gnecco-heredia@dauphine.psl.eu - C602

* *Remerciement spécial à Florian Yger*

Qualité attendue d'un classifieur

- précision : taux d'erreur (proportion d'individus mal classés) aussi bas que possible
- robustesse : le modèle ne doit pas trop dépendre de l'échantillon d'apprentissage (i.e. après un autre échantillonnage, la décision sera similaire) et généraliser à d'autres données
- concision/parcimonie : le modèle doit faire intervenir peu de paramètres
- rapidité de calcul : pour l'entrainement et la prédiction

Compromis complexité / généralisation

Illustration

Compromis complexité / généralisation

Illustration

Comment évaluer la capacité de généralisation d'un modèle ?

Sur/sous apprentissage

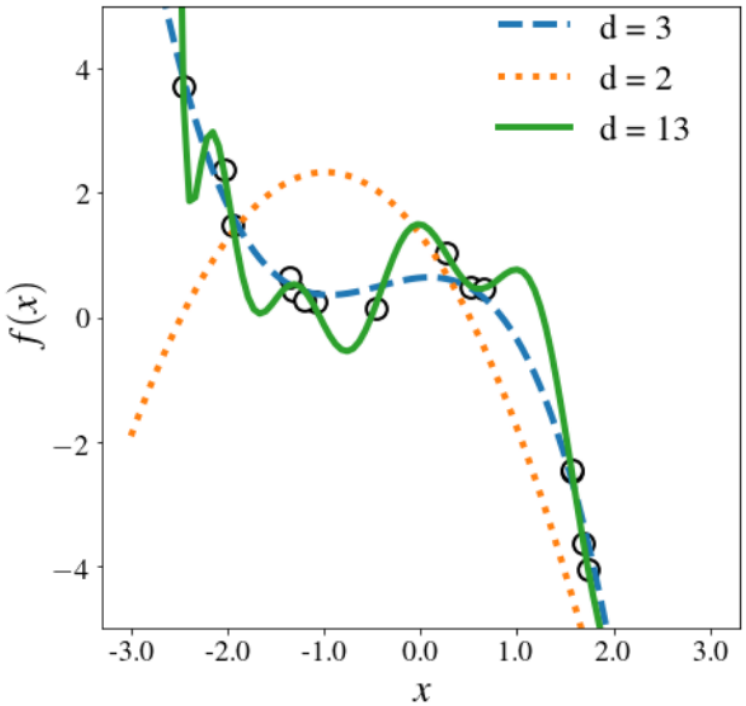


Figure – [Has+09]

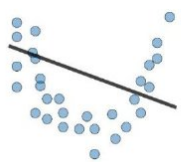
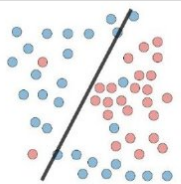
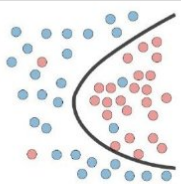
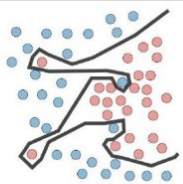
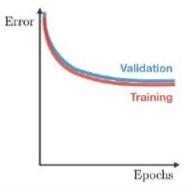
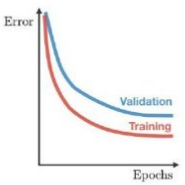
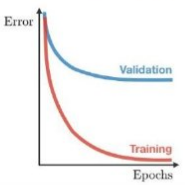
	Underfitting	Just right	Overfitting
Symptoms	- High training error - Training error close to test error - High bias	- Training error slightly lower than test error	- Low training error - Training error much lower than test error - High variance
Regression			
Classification			
Deep learning			
Remedies	- Complexify model - Add more features - Train longer		- Regularize - Get more data

Figure – Taken from here

Matrice de confusion

	Prédiction		
Donnée		Yes / 1	No /0
	Yes / 1	TP	FN
	No / 0	FP	TN

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

Matrice de confusion

		Prédiction	
		Yes / 1	No /0
Donnée	Yes / 1	TP	FN
	No / 0	FP	TN

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

Matrice de confusion

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

Suivant les applications

On pourra donner une importance différente à ces quantités

Exemple

Dans certaines applications médicales, un FP peut être moins "grave" qu'un FN

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
  $\text{Acc} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} = 1 - \text{Err}$
  limite : dépend fortement des probabilités a priori des classes

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)
  $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$
  Parmi les exemples positifs, combien sont détectés.
  On a aussi $\text{TNR} = \frac{\text{TN}}{\text{TN+FP}}$

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)
- Précision (*Precision* / PPV - *Positive Predicted Values*)

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)
- Précision (*Precision* / PPV - *Positive Predicted Values*)
  $\text{PPV} = \frac{\text{TP}}{\text{TP}+\text{FP}}$
  Parmi les exemples positifs détectés par le classifieur, combien sont correctes.
  Précision et Rappel sont antinomiques.

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)
- Précision (*Precision* / PPV - *Positive Predicted Values*)
- F-mesure

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)
- Précision (*Precision* / PPV - *Positive Predicted Values*)
- F-mesure

  Moyenne harmonique de la Précision et du Rappel

  $\mathrm{F1} = 2\frac{\mathrm{PPV}\cdot\mathrm{TPR}}{\mathrm{PPV}+\mathrm{TPR}} = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}$

Métriques

Ingrédient de base

- TP - True Positive
- FP - False Positive
- TN - True Negative
- FN - False Negative

- Taux de bonne classification (*Accuracy*)
- Rappel (*Recall* / TPR - *True Positive Rate*)
- Précision (*Precision* / PPV - *Positive Predicted Values*)
- F-mesure

Pour chacune de ces métriques, l'idéal serait d'être maximal.

Métriques

Ingrédient de base

- TPR - True Positive Rate
  $\text{TPR} = \frac{\text{TP}}{\text{TP+FN}}$
- PPV - Positive Predicted Values
  $\text{PPV} = \frac{\text{TP}}{\text{TP+FP}}$

- Courbe ROC (*Receiver Operating Characteristic*)
  Cette courbe représente l'évolution du TPR en fonction du FPR en faisant varier un seuil de classification (ex : $P_0(x) < \theta$)
  Ex : $\hat{y}_i = 1$ si $\hat{P}(y = 1|x_i) > \theta$, on calcule les TPR et FPR de cette règle et on fait varier θ

Illustration

Métriques

- Courbe ROC (*Receiver Operating Characteristic*)
  Cette courbe représente l'évolution du TPR en fonction du FPR en faisant varier un seuil de classification (ex : $P_0(x) < \theta$)
  Ex : $\hat{y}_i = 1$ si $\hat{P}(y = 1|x_i) > \theta$, on calcule les TPR et FPR de cette règle et on fait varier θ

  Illustration

- Aire sous la courbe ROC (AUC - *Area Under the roc Curve)* donne une notion de la façon dont le classifieur ordonne les exemples (*ranking*)
  Une décision aléatoire aura une AUC ≈ 0.5 et le classifieur parfait aura une AUC de 1.

Métriques

- Courbe ROC (*Receiver Operating Characteristic*)
  Cette courbe représente l'évolution du TPR en fonction du FPR en faisant varier un seuil de classification (ex : $P_0(x) < \theta$)
  Ex : $\hat{y}_i = 1$ si $\hat{P}(y = 1|x_i) > \theta$, on calcule les TPR et FPR de cette règle et on fait varier θ

  Illustration

- Aire sous la courbe ROC (AUC - *Area Under the roc Curve)*
  donne une notion de la façon dont le classifieur ordonne les exemples (*ranking*)
  Une décision aléatoire aura une AUC ≈ 0.5 et le classifieur parfait aura une AUC de 1.

Pour chacune de ces métriques, l'idéal serait d'être maximal.

Quel modèle utiliser ?

Exemple

Pour un jeu de données $\mathcal{D}$, on se demande :

- Quel classifieur utiliser entre LDA, QDA et régression logistique ?
- Quel choix d'hyperparamètre pour un k-ppv ? ($k = 1$ vs $k = 10$ ...)

Problématique

On va chercher à évaluer les performances des modèles (à la vue d'une des métriques précédentes) en généralisation.

Méthode naïve I

- entrainer chaque classifieur sur l'ensemble d'apprentissage
- vérifier leurs performances sur ce même ensemble

Mais...

- On n'évalue pas la capacité de généralisation du modèle...
- plutôt l'apprentissage "par cœur" du modèle.
- Méthode trop optimiste

Méthode naïve II

- entrainer chaque classifieur sur l'ensemble d'apprentissage
- vérifier leurs performances sur un ensemble de test

Remarque

- nécessite de séparer le jeu de données en deux ensembles indépendants (en général $80\% - 20\%$ ou $90\% - 10\%$ app/test)
- cet échantillonnage aléatoire doit maintenir les probabilités a priori (stratification)
- **sensibilité à l'échantillonnage**

Validation croisée

en anglais : *Cross-validation*

- validation croisée aléatoire

Validation croisée

- validation croisée aléatoire :
  - sélectionner T observations parmi les données (sans remise) pour l'apprentissage et tester sur le reste
  - répéter la procédure et faire la moyenne/variance des résultats obtenus par chaque modèle

Validation croisée

- validation croisée aléatoire
- validation croisée à K-blocs (*K-fold Cross Validation*)

Illustration

  - découper le jeu de données en K ensembles disjoints de $\frac{n}{K}$ observations
  - utiliser à chaque fois un des ensembles comme ensemble de test et effectuer l'apprentissage sur les $K-1$ ensembles
  - évaluer les performances sur la moyenne (et la variance) de ces K expériences

Validation croisée

- validation croisée aléatoire
- validation croisée à K-blocs (*K-fold Cross Validation*)
- validation croisée à n-blocs (*Leave-One-Out Cross Validation*)
  - cas extrême de validation croisée à K-blocs
  - couteux en temps de calcul
  - approxime le mieux la généralisation

Validation croisée

- validation croisée aléatoire
- validation croisée à K-blocs (*K-fold Cross Validation*)
- validation croisée à n-blocs (*Leave-One-Out Cross Validation*)

Détails importants

- stratification
- test statistique apparié (test de student, test du signe de Cox-Wilcoxon)
- fixer la graine du générateur pseudo-aléatoire (reproductibilité)

La validation croisée est la clé dans de nombreuses applications.

Cas multi-classe

État des lieux

Certaines méthodes sont nativement déjà adaptées au cas multi-classe ou bien possède des variantes pour ce cas

- k-ppv
- régression logistique mutlinomiale

Cas multi-classe

État des lieux

Certaines méthodes sont nativement déjà adaptées au cas multi-classe ou bien possède des variantes pour ce cas

- k-ppv
- régression logistique mutlinomiale

Il est cependant possible de transformer un classifieur binaire en classifieur multi-classe

Stratégie un contre un

Pour un jeu de données contenant K classes

- entraînement du classifieur sur chaque paire de classes possible dans le jeu de données (1 vs 2, 1 vs 3, ..., (K-1) vs K))
- on genère donc $K * (K-1)/2$ modèles de prédiction
- pour la prédiction d'une nouvelle observation, chaque modèle prédit une classe et on choisit la classe majoritairement prédite

Remarques

- en anglais *one vs one*
- heuristique très coûteuse (on doit lancer $K * (K-1)/2$ apprentissages -même si on utilise à chaque fois un sous-ensemble des données).

Stratégie un contre tous

Pour un jeu de données contenant K classes

- entraînement du classifieur sur une classe (choisie comme la classe positive) et le reste constituera la classe négative (1 vs (2,...,K), K vs (1,..., K-1))
- on genère donc K modèles de prédiction
- pour la prédiction d'une nouvelle observation, chaque modèle doit pouvoir prédire un score (proba par ex.) pour chaque classe et on choisit la classe ayant le plus grand score.

Remarques

- en anglais *one vs all* ou *one vs rest*
- heuristique moins coûteuse (on ne lance "que" K apprentissages).
- limitée aux approches permettant d'évaluer un score

Exemple

Le jeu de données contient 4 classes (A, B, C, D)

Un contre un

- on entraîne 6 modèles
- pour la prédiction de x, on obtient
  A vs B prédit A
  A vs C prédit A
  A vs D prédit A
  B vs C prédit B
  B vs D prédit D
  C vs D prédit D
- on assigne alors x à la classe A

Exemple

Le jeu de données contient 4 classes (A, B, C, D)

Un contre tous

- on entraîne des modèles A vs All, B vs All,C vs All et D vs All
- pour la prédiction de x, on obtient les scores
  A vs All $= 40\%$
  B vs All $= 30\%$
  C vs All $= 60\%$
  D vs All $= 50\%$
- on assigne alors x à la classe C