SD210 - Bases de l'apprentissage statistique
Cours 2 - Partie 2 Sélection et évaluation de modèles

Florence d'Alché–Buc
Institut Mines-Télécom, Télécom ParisTech, LTCI umr CNRS 5141
florence.dalche@telecom-paristech.fr

Outline

Rappel: apprentissage supervisé par approche régularisée

Construire une fonction $\hat{h}$ un classifieur de $\mathbb{R}^p$ vers $\{-1, 1\}$ (resp. $\{1 \ldots, C\}$) telle que:

$$\hat{h} = \arg\min_{h \in \mathcal{H}} \sum_{i=1}^{n} L(y_i, h(x_i)) + \lambda \Omega(h) \tag{1}$$

- L est une **fonction de perte locale**: L mesure à quel point $h(x_i)$ est proche de y_i la valeur de sortie désirée.
- $\sum_{i=1}^{n} L(y_i, h(x_i))$: terme d'attache aux données
- $\Omega(h)$: pénalité sur la fonction h, contrôle la complexité du modèle, par exemple la norme au carré du vecteur de paramètres de h.

NB: l'approche régularisée n'est pas la seule approche possible, mais c'est la plus courante !

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des données

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres

Résoudre un problème d'apprentissage d'un classifieur

Méthodologie pour développer une approche discriminante

- Définir
  - l'**espace de représentation** des données
  - la **classe des fonctions** de classification binaire considérées
  - la **fonction de coût** à minimiser pour obtenir le meilleur classifieur dans cette classe
  - l'**algorithme de minimisation** de cette fonction de coût
  - une **méthode de sélection de modèle** pour définir les hyperparamètres
  - une méthode d'évaluation des performances

Sélection ou évaluation de modèle ?

- Estimer les performances de différents modèles afin de choisir le meilleur hyperparamètre : sélection de modèle
- Ayant appris un modèle étant donnés ses hyperparamètres, estimer ses performances: évaluation de modèle

Aujourd'hui, nous nous concentrons sur ces deux questions.
N.B.: On choisit le même critère : erreur de prédiction, aire sous la courbe ROC pour les deux étapes

Premier exemple: sélection de modèles

Un classifier linéaire dans le plan:

$$h(x) = \text{signe}(\beta_1 x_1 + \beta_2 x_2 + \beta_0) \tag{2}$$

Apprendre h_β en minimisant: $\sum_{i=1}^{n} L(y_i, h(x_i)) + \lambda \|\beta\|_2^2$
OU $\sum_{i=1}^{n} L(y_i, h(x_i)) + \lambda \|\beta\|_1$ Quelle valeur de λ choisir ?

Deuxieme exemple: le paramètre de lissage K

Comment choisir *K* ? K: trop petit : la fonction *f* est trop sensible aux données : large variance
K : trop large : la fonction *f* devient trop peu sensible aux données : biais important

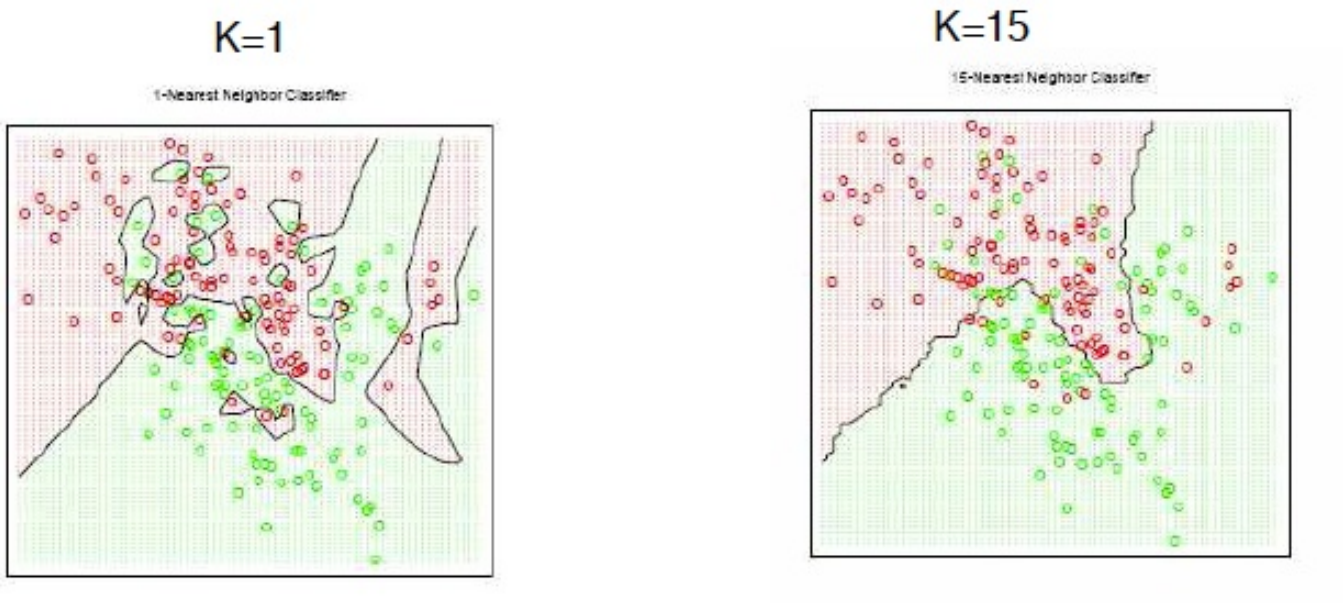


Fig 2.2, 2.3 of HTF01

Book of Hastie, Tibshirani and Friedman (The elements of statistical learning, Springer)

Décomposition biais-variance

On suppose: $Y = f(X) + \varepsilon$ avec ε centré et de variance σ_ε^2. Soit $\hat{f}$, la fonction apprise à partir de $\mathcal{S}$, l'échantillon d'apprentissage.
Erreur quadratique (régression) espérée aprés apprentissage sur un échantillon de taille n: $\mathcal{S}$

$$\mathbb{E}_{X,Y,\mathcal{S}}[(Y - \hat{f}(X))^2] = \mathbb{E}_X \mathbb{E}_{Y|X,\mathcal{S}}[(Y - \hat{f}(X))^2]$$

Soit:

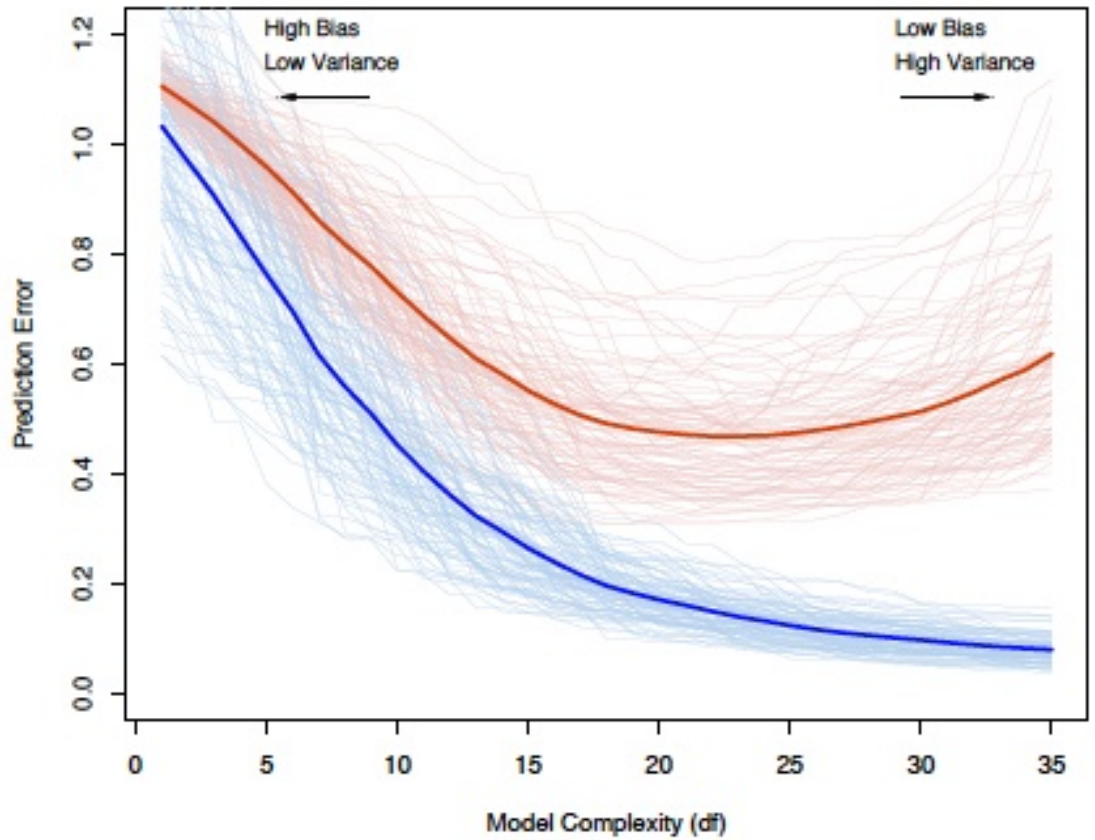
$$\mathbb{E}_X \left[\mathbb{E}_{Y|X,\mathcal{S}}(Y - f(X))^2 + \mathbb{E}_{Y|X,\mathcal{S}}[(\bar{f}(X) - \hat{f}(X))^2] + \mathbb{E}_{Y|X}[(f(X) - \bar{f})^2] \right]$$

$$\text{soit } \sigma_\varepsilon^2 + \mathbb{E}_X[Var_{\mathcal{S}}[\hat{f}(X)] + \text{Biais}_{\mathcal{S}}^2(\hat{f}(X))]]$$

Avec $\bar{f}(X) = \mathbb{E}_{\mathcal{S}} \hat{f}(X)$
Terme incompressible : bruit des données
Biais au carré: mesure à quel point $\hat{f}$ est loin de la cible
Variance de $\hat{f}(X)$: mesure à quel point $\hat{f}$ est sensible aux données d'apprentissage

Biais variance

Faisons varier $\mathcal{S}$ léchantillon d'apprentissage.



Book of Hastie, Tibshirani and Friedman (The elements of statistical learning, Springer)

Décomposition biais-variance des k-plus-proches voisins

Posons $X = x_0$ donc l'aléa ne vient pas de x_0. On a :

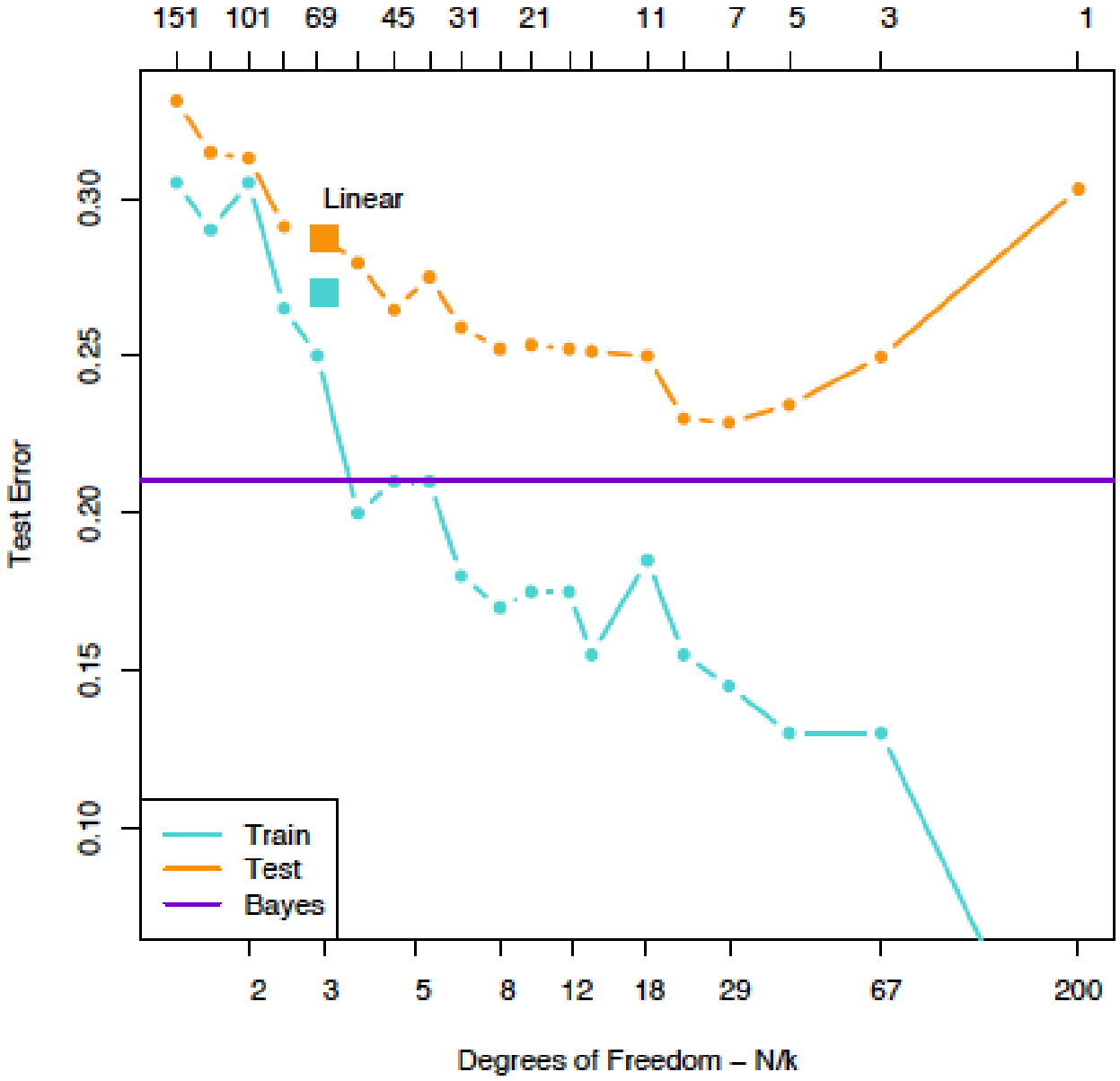
$$\mathbb{E}[(Y - \hat{f}(x_0))^2] \quad = \quad Var[Y] + \text{Biais}^2[f(x_0)] + Var(\hat{f}(x_0))$$

En plus on fixe $\mathcal{S}$, donc $\hat{f}(x_0)$ est déterministe :

$$\mathbb{E}[(Y - \hat{f}(x_0))^2] = \sigma_\varepsilon^2 + (f(x_0) - \frac{1}{K}\sum_{\ell=1}^{K} f(x_{(\ell)})^2 + \frac{\sigma_\varepsilon^2}{K}$$

K contrôle le terme de variance : plus la valeur de K est grande, plus la variance décroit; mais K contrôle aussi le biais, plus la valeur de est petite, plus petit est le biais : nous sommes en plein *dilemne biais-variance*. Donc choisir K est primordial !

Erreur de test en fonction de $\frac{n}{\nu}$



Book of Hastie, Tibshirani and Friedman (The elements of statistical learning,

k-nn

Le classifieur des k-plus-proches voisins: classifieur paresseux : pas besoin d'algorithme d'apprentissage ! J'ai besoin des données dites d'apprentissage, d'une métrique et de la valeur de K

- Comment choisir la valeur de K ?
- Ayant choisi $\widetilde{K}$, comment estimer l'erreur en généralisation de ce K-NN ?

Sélection de modèle

Validation croisée :

- Partager l'échantillon disponible en trois sous-échantillons: Apprentissage , Validation , Test
- On effectue une procédure de validation croisée sur l'ensemble dit de validation : cette procédure nous fournit la valeur du meilleur hyperparamètre pris dans une grille finie. On apprend sur Apprentissage avec cet hyperparamètre. Puis on teste la fonction obtenue sur l'ensemble test

Validation croisée

1. Diviser les données S en B parties de même taille (approximativement) et disjointes $S_{b=1}, \ldots, D_{b=B}$.
2. Pour $b \in \{1, \ldots, B\}$:
   - Entrainer le modèle $\mathcal{H}_\lambda$ sur toutes les données **sauf** D_b pour obtenir un estimateur $\hat{h}^b_{\lambda,n}$
   - Calculer **sur les données restantes** D_b (test) le risque empirique

$$R_{n,b}(\lambda) = \frac{1}{|D_b|} \sum_{j \in D_b} L(x_j, y_j, \hat{h}^b_{\lambda,n})$$

3. Risque estimé par validation croisée

$$R^B_{n,CV}(\lambda) = \frac{1}{B} \sum_{b=1}^{B} R_{n,b}(\lambda)$$

Trouver λ

Répéter cette procédure sur tous les $\lambda \in \Lambda$ considérés et choisir

$$\hat{\lambda}_{n,B} = \arg\min_{\lambda \in \Lambda} R_{n,CV}(\lambda). \tag{3}$$

Sélection de modèles

- On sélectionne sur $\mathcal{S}_{val}$:$\widetilde{\lambda}$
- On apprend sur $\mathcal{S}_{app}$ en utilisant $\widetilde{\lambda}$
- On teste sur $\mathcal{S}_{test}$ en utilisant la fonction apprise à partir de $\mathcal{S}_{app}$ et $\widetilde{\lambda}$

$R_{CV,val}$ nous dit à quel type d'erreur en généralisation nous attendre en apprenant sur un ensemble de taille $n_{val} - n_{val}/B$. $R_{\mathcal{S}_{app}}$ nous dit à quel point le classifieur a bien réussi à approcher les données d'apprentissage
$R_{\mathcal{S}_{test}}$ nous dit à quel point le classifieur réussit à approcher les données (nouvelles) de test

Outline

Critère d'évaluation des modèles

On choisit en général:

- l'erreur quadratique pour la régression
- l'erreur de prédiction (0-1) pour la classification

Cependant pour la classification binaire, on souhaite souvent avoir plus de détails sur la nature des erreurs: faux positif, faux négatif... on se tourne donc vers les courbes ROC et l'aire sous la courbe ROC.

Courbes ROC

	Prédit OUI	Prédit NON
POS	Vrai positifs	Faux négatifs
NEG	Faux positifs	Vrais négatifs

TPR : taux de faux positifs, $TPR = \frac{TP}{TP+FN} = \frac{nb\ pos\ oui}{nb\ pos}$

FPR : taux de faux positfs, $FPR = \frac{FP}{FP+TN} = \frac{nb\ oui\ neg}{nb\ neg}$

Courbes ROC

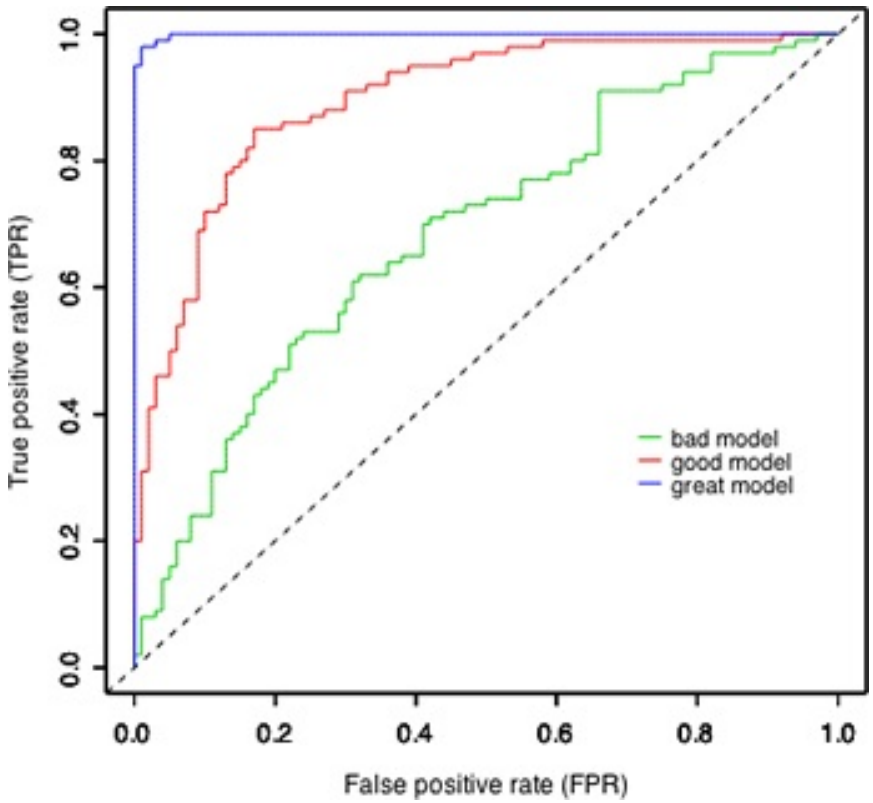
Soit h un classifieur défini par: $h(x) = \text{signe}(f(x) - s)$
Habituellement en classification biclasse, $s = 0.5$ si on approche des sorties $\{0, 1\}$
ou bien $s = 0$ ou $\{-1, 1\}$

Définition

Tracer une courbe ROC consiste à faire varier le seuil s et à chaque fois à reporter le point (taux de faux positifs, taux de vrai positifs) sur un graphique 2D. Ainsi à une même fonction f issue d'un algorithme d'apprentissage on associe plusieurs points, autant que de seuils $s_1, \ldots, s_m$ qu'on considère. NB: la courbe est continue lorsque le classifieur est probabiliste.

Comparer deux classifieurs avec une courbe ROC

Sur un ensemble test, pour une fonction f donnée, je mesure (FPR(s),TPR(s)) en faisant varier s.



Aire sous la courbe ROC

L' aire sous la courbe ROC (valant donc entre 0 et 1) est un bon indicateur de la qualité d'un classifieur et permet également de comparer de manière plus juste (en tenant compte de l'ensemble des seuils) deux classifieurs.

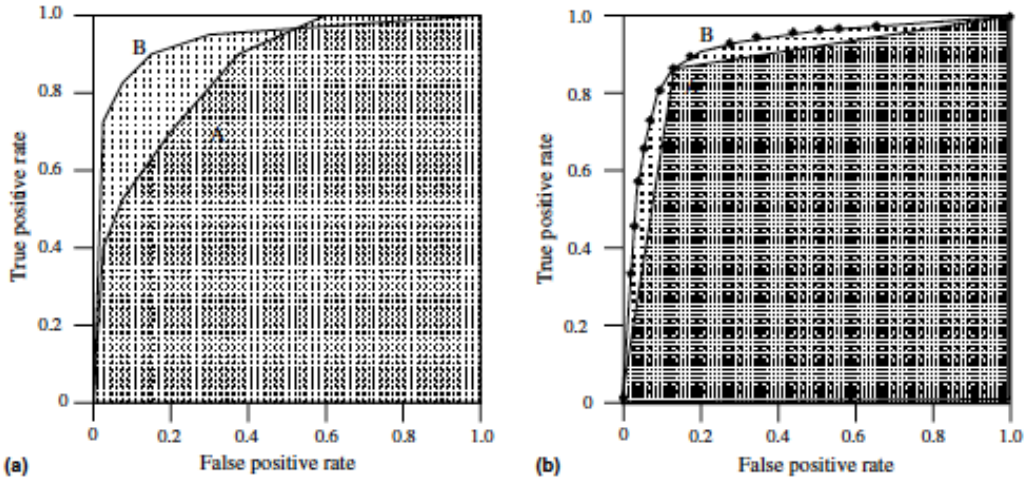


Fig. 8. Two ROC graphs. The graph on the left shows the area under two ROC curves. The graph on the right shows the area under the curves of a discrete classifier (A) and a probabilistic classifier (B).

Réf: pour aller plus loin, lire l'article de Fawcett (2005): introduction to ROC analysis (disponible sur le site pédagogique de SD210)

Pour aller plus loin ... D'autres courbes

En recherche d'information, on cherche à savoir si ce qu'on prédit (par exemple par l'intermédiaire d'un moteur de recherche) est bien de la classe 1, on trace alors plutôt des courbes Précision-Rappel avec les définitions suivantes:

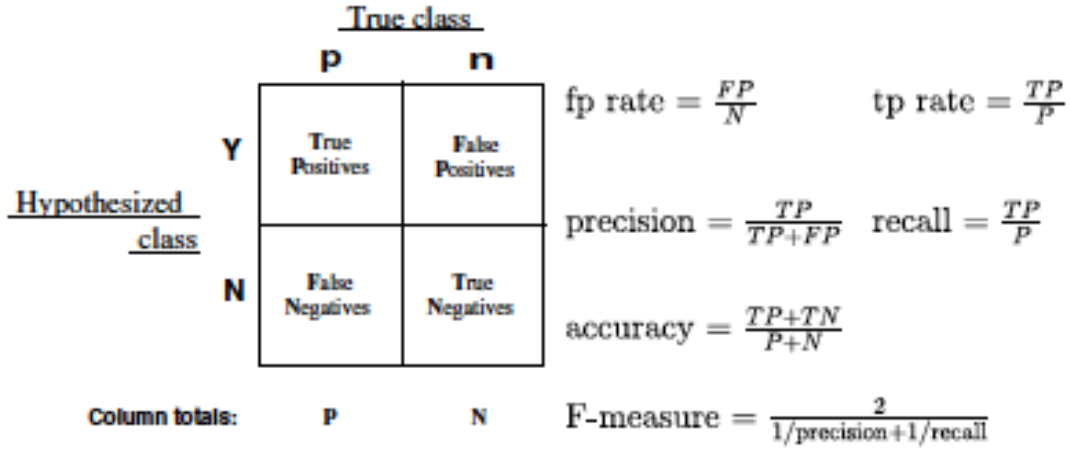


Fig. 1. Confusion matrix and common performance metrics calculated from it.

Réf: lire l'article de Fawcett (2005): introduction to ROC analysis (disponible sur le site pédagogique de SD210)

Exemple de courbes ROC et PR

Les courbes ROC sont insensibles au déséquilibre de classes alors que les courbes Precision-Rappel le sont: voici deux exemples de courbes ROC et PR tracées respectivement sur des classes équilibrées puis déséquilibrées.

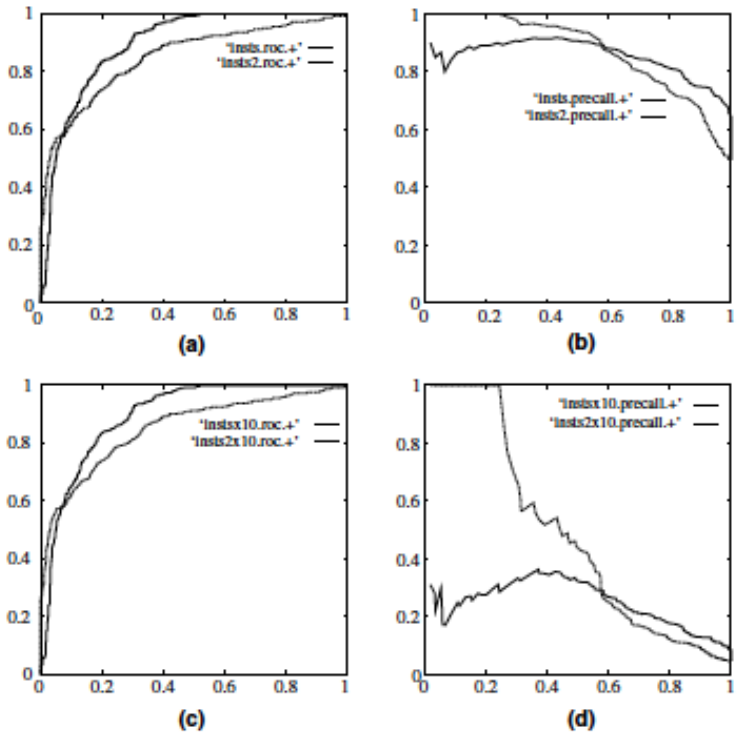


Fig. 5. ROC and precision-recall curves under class skew. (a) ROC curves, 1:1; (b) precision-recall curves, 1:1; (c) ROC curves, 1:10 and (d) precision-recall curves, 1:10.

Réf: lire l'article de Fawcett (2005): introduction to ROC analysis

Références

- Chapitres 4 et 7 de Elements of statistical learning, HTF.
- T. Fawcett, Introduction to ROC analysis, Pattern Recognition Letters, 27:861-874, 2006.