Introduction à l'Extraction de Connaissances

Chapitre IV : les méthodes

Part 3

Introduction à l'évaluation

Alexandre Saidi
Master -Informatique
ECL - LIRIS - CNRS

Novembre 2017

Décomposition Biais et Variance

Rappels et notations

- Pour une base de données (X, Y), on note $Y = f(X) + \varepsilon$
  - ➛ Y l'ensemble des classes (observées), X les observations (explicatives) et
  - ➛ ε une erreur aléatoire indépendante de X avec une moyenne de zéro
- On dira que f représente l'information *systématique* que X donne sur Y.
- Ne pas confondre ε avec l'erreur de la prédiction.
  - ε peut venir des variables non mesurées (cachées) utiles pour calculer Y.
  - Elle peut également venir des variations non mesurées / le bruit / ...
  - On ne pourra pas réduire l'erreur ε quelque soit la qualité de notre apprentissage
    - ➛ Car Y est (aussi) une fonction de ε.
- On note la prédiction $\hat{Y} = \hat{f}(X)$
- Si on suppose que $\hat{f}$ et X sont fixes, la valeur attendue (l'espérance / la moyenne) de la différence entre la **réalité** de la BD (Y) et la **prédiction** $(\hat{Y})$:

$$\mathbb{E}(Y - \hat{Y})^2 = \mathbb{E}[f(X) + \varepsilon - \hat{f}(X)]^2 = \mathbb{E}[f(X) - \hat{f}(X)]^2 + Var(\varepsilon)$$

➛ le terme $[f(X) - \hat{f}(X)]^2$ est réductible mais pas $Var(\varepsilon)$

- ε donnera toujours une borne supérieure de la justesse du prédicteur.

Décomposition Biais et Variance (suite)

D'où vient $Var(\varepsilon)$?

• Notons d'abord que l'estimation qui minimise l'erreur de prédiction (soit $\mathbb{E}(y-\hat{y})^2$ notée $\mathbb{E}_y(y-\hat{y})^2$ pour plus de précision) est donnée par :

$$\frac{\partial}{\partial \hat{y}} \mathbb{E}_y\{(y-\hat{y})^2\} = 0$$

➛ $\mathbb{E}_y\{-2(y-\hat{y})\} = 0$ simple dérivée p/r $\hat{y}$

➛ $\mathbb{E}_y\{y\} - \mathbb{E}_y\{\hat{y}\} = 0$

➛ $\mathbb{E}_y\{y\} = \mathbb{E}_y\{\hat{y}\}$

L'estimation qui minimise l'erreur est donc $\mathbb{E}_y\{y\}$ (appelée modèle de Bayes)

☞ Dans la pratique, on ne peut pas calculer la valeur exacte de $\mathbb{E}_y\{y\}$ à moins d'avoir TOUTES les valeurs de y auquel cas il n'y a plus besoin de prédiction !

• Nous estimons y en l'absence de la probabilité $Pr(y)$ en utilisant un ensemble d'apprentissage $LS = \{y_1, y_2, ..., y_n\}$

P. Ex., s'il s'agit de la taille des Suédois (y), on notera une deux estimations :

$\hat{y_1} = \frac{1}{n}\sum_{i=1}^{n} y_i$ ou

$\hat{y_2} = \frac{250\lambda + \sum_{i=1}^{n} y_i}{\lambda + n}$, $\lambda \in [0, \infty[$ si on sait la taille proche de 2,50m.

Décomposition Biais et Variance (suite)

• On demande à un bon modèle de ne pas être juste bon sur un ensemble d'apprentissage mais sur tous (Ls).

$$\mathbb{E}_y(y-\hat{y})^2 = \mathbb{E}_{\mathbb{LS}}\{\mathbb{E}_y\{(y-\hat{y})^2\}\}$$

D'où

$$\begin{aligned}
&\mathbb{E}_{\mathbb{LS}}\{\mathbb{E}_y\{(y-\hat{y})^2\}\} \\
&= \mathbb{E}_{\mathbb{LS}}\{\mathbb{E}_y\{(y-\mathbb{E}_y\{y\}+\mathbb{E}_y\{y\}-\hat{y})^2\}\} && \text{On ajoute et enlève} \\
&= \mathbb{E}_{\mathbb{LS}}\{\mathbb{E}_y\{(y-\mathbb{E}_y\{y\})^2\}\} + \mathbb{E}_{\mathbb{LS}}\{(\mathbb{E}_y\{(\mathbb{E}_y\{y\}-\hat{y})^2\}\} \\
&\quad +\mathbb{E}_{\mathbb{LS}}\{\mathbb{E}_y\{2(y-\mathbb{E}_y\{y\})(\mathbb{E}_y\{y\}-\hat{y}))\}\} && \text{on aura un terme nul} \\
&= \mathbb{E}_y\{(y-\mathbb{E}_y\{y\})^2\} + \mathbb{E}_{\mathbb{LS}}\{(\mathbb{E}_y\{y\}-\hat{y})^2\} + \mathbb{E}_{\mathbb{LS}}\{2(\mathbb{E}_y\{y\}-\mathbb{E}_y\{y\})(\mathbb{E}_y\{y\}-\hat{y})\} \\
&= \mathbb{E}_y\{(y-\mathbb{E}_y\{y\})^2\} + \mathbb{E}_{\mathbb{LS}}\{(\mathbb{E}_y\{y\}-\hat{y})^2\}
\end{aligned}$$

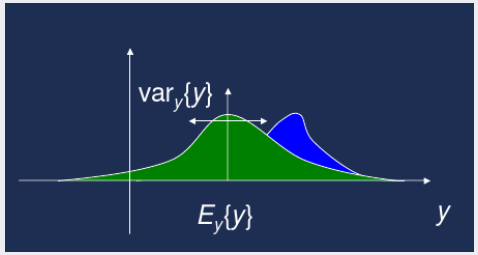
• On a donc l'expression de l'erreur (*de test*) E :

$$E = \mathbb{E}_y\{(y-\mathbb{E}_y\{y\})^2\} + \mathbb{E}_{\mathbb{LS}}\{(\mathbb{E}_y\{y\}-\hat{y})^2\}$$

Le premier terme est l'erreur résiduelle

$= var_y\{y\} = var(\varepsilon)$

qui est le minimum atteignable de (toute) l'erreur E



• On développe le 2e terme ci dessus : $\mathbb{E}_{\mathbb{LS}}\{(\mathbb{E}_y\{y\}-\hat{y})^2\}$ qui nous donnera une expression du biais et de la variance de l'erreur d'apprentissage.

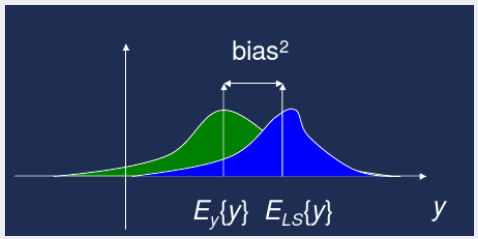
Décomposition Biais et Variance (suite)

Développement du (2e) terme $\mathbb{E}_{LS}\{(\mathbb{E}_y\{y\} - \hat{y})^2\}$

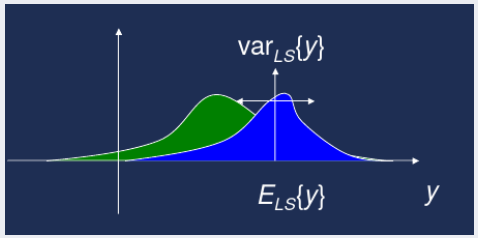
$$
\begin{aligned}
& \mathbb{E}_{LS}\{(\mathbb{E}_y\{y\} - \hat{y})^2\} \\
= \; & \mathbb{E}_{LS}\{(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\} + \mathbb{E}_{LS}\{\hat{y}\} - \hat{y})^2\} \\
= \; & \mathbb{E}_{LS}(\{(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\})^2\} + \mathbb{E}_{LS}\{(\mathbb{E}_{LS}\{\hat{y}\} - \hat{y})^2\} \\
& + \mathbb{E}_{LS}\{2(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\})(\mathbb{E}_{LS}\{\hat{y}\} - \hat{y})\}\} \\
= \; & (\{(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\})^2 + \mathbb{E}_{LS}\{(\hat{y} - \mathbb{E}_{LS}\hat{y})^2\} \\
& + 2(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\})(\mathbb{E}_{LS}\{(\mathbb{E}_{LS}\{\hat{y}\} - \hat{y})\}\} \\
= \; & (\{(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\})^2 + \mathbb{E}_{LS}\{(\hat{y} - \mathbb{E}_{LS}\{\hat{y}\})^2\}
\end{aligned}
$$

terme nul

- On a le 1er terme $(\{(\mathbb{E}_y\{y\} - \mathbb{E}_{LS}\{\hat{y}\})^2$
  où $\mathbb{E}_{LS}\{\hat{y}\}$ = le modèle moyen (construit sur tous les ensembles L_S)
  $= biais^2$ = erreur entre le modèle Bayes (ci-dessus) et le modèle moyen.

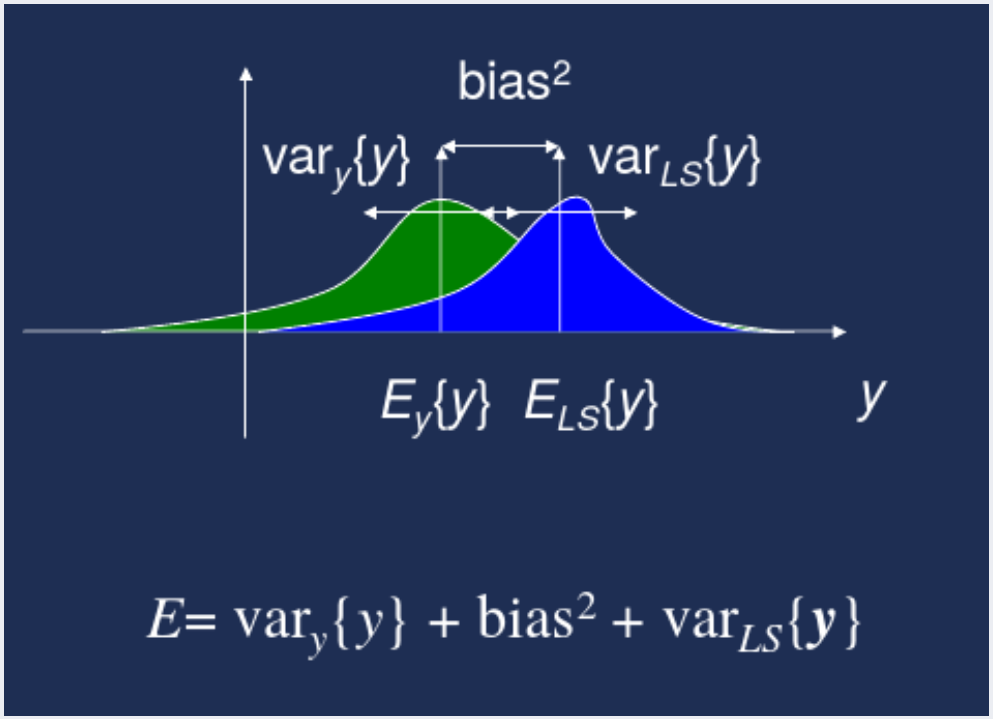


- Enfin, le 2e terme $\mathbb{E}_{LS}\{(\hat{y} - \mathbb{E}_{LS}\{\hat{y}\})^2\} = var_{LS}\{y\}$
  = une estimation de la variance
  = une mesure de l'overfitting éventuel.



Décomposition Biais et Variance (suite)

- Par conséquent : $E = \varepsilon + biais^2 + var_{LS}(y)$



Notons : la variance $\varepsilon = var_y(y)$ est le **bruit** (noise, Vars. cachées, ...),
$biais^2$ est l'erreur entre l'estimation et les observations (la B.D.)
$var_{LS}(y)$ est la variance de l'estimation-même (app. faits sur $\neq$ BDs.)

Décomposition Biais et Variance (suite)

Rappel : on avait deux estimations de la **taille des Suédois** (y) :

$\hat{y_1} = \frac{1}{n}\sum_{i=1}^{n} y_i$ ou

$\hat{y_2} = \frac{250\lambda + \sum_{i=1}^{n} y_i}{\lambda + n}$, $\lambda \in [0, \infty[$ si on sait la taille proche de 2,50m.

- 1- Pour le premier cas, on obtient :

  $biais^2 = (\{(\mathbb{E}_y\{y\} - \mathbb{E}_{\mathbb{LS}}\{\hat{y_1}\})^2 = 0$ 2 termes identiques

  $var_{LS}(\hat{y_1}) = \frac{1}{n} var_y\{y\}$

  ➙ C-à-d, du point de vu statistique, $\hat{y_1}$ **est la meilleure estimation** avec un biais nul.

- 2- Pour le deuxième cas, on obtient :

  $biais^2 = \frac{\lambda}{\lambda+n}((\mathbb{E}_y\{y\} - 250)^2$ avec $\lambda \in [0, \infty[$ les tailles sont proches de 2,50m.

  $var_{LS}(\hat{y_2}) = \frac{n}{(\lambda + n)^2}\ var_y\{y\}$

  La variance obtenue dépend de la variance $\varepsilon = var_y\{y\}$ et ne permet pas de dire que l'estimateur obtenu $\hat{y_2}$ est le meilleur.

  ➙ Ici, il y a le dilemme biais - variance .

Décomposition Biais et Variance (suite)

Pourquoi $\hat{y_2} = \dfrac{250\lambda + \sum_{i=1}^{n} y_i}{\lambda + n},\ \lambda \in [0, \infty[$ la taille proche de 2,50m.

- On fait 2 hypothèses :
  - La moyenne de la taille est proche de 2m50 :

$$Pr(\overline{y}) = A_{\sigma_{\overline{y}}}\ exp\left(-\frac{(\overline{y} - 250)^2}{2\sigma_{\overline{y}}^2}\right)$$ A est une constante (voir ci-après)

  - La taille d'un individu est proche de la moyenne gaussienne :

$$Pr(y_i|\overline{y}) = B_{\sigma}\ exp\left(-\frac{(y - \overline{y})^2}{2\sigma_{y}^2}\right)$$ Idem (voir ci-après)

- La valeur **la plus probable** de $\overline{y}$ sachant l'ensemble d'apprentissage L_S sera :

$$\begin{aligned}
\hat{y} &= argmax_{\overline{y}}\ Pr(\overline{y}|L_S) && \text{on multiplie par } Pr(L_S) \text{ qui est cste,}\\
&= argmax_{\overline{y}}\ Pr(L_S|\overline{y})Pr(\overline{y}) && \text{puis on utilise le théorème de Bayes}\\
&= argmax_{\overline{y}}\ Pr(y_1, y_2, ..., y_n|\overline{y})Pr(\overline{y})\\
&= argmax_{\overline{y}}\ \prod_{i=1}^{n} Pr(y_i|\overline{y})Pr(\overline{y}) && \text{Indép. des variables}\\
&= argmin_{\overline{y}}\ -\sum_{i=1}^{n} log(Pr(y_i|\overline{y})) - log()Pr(\overline{y})) && ../..
\end{aligned}$$

Décomposition Biais et Variance (suite)

Et par les 2 hypothèses ci-dessus, on obtient :

$$= argmin_{\overline{y}} \sum_{i=1}^{n} \frac{(y-\overline{y})^2}{2\sigma_y^2} + \frac{(\overline{y}-250)^2}{2\sigma_{\overline{y}}^2}$$

$$= \quad \ldots.$$

$$= \frac{250\lambda + \sum_{i=1}^{n} y_i}{\lambda + n} \text{ avec } \lambda = \frac{\sigma_y^2}{\sigma_{\overline{y}}^2}$$

➙ L'approche Bayésienne permet d'expliquer et de calculer λ

Illustrations

Rappels et motivations :

• Pourquoi faut-il tant de méthodes au lieu de la "meilleure" ?
➙ Il n'y a pas de meilleure méthode qui s'appliquerait dans tous les cas !

• Il est donc important de connaître la meilleure méthode pour une BD donnée.
➙ Mais comment choisir la meilleure méthode ?

Exemple d'erreur MSE

• Une mesure de l'erreur très utilisée : MSE (*mean squared erreur* : moyenne de l'erreur au carrée)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{f}(x_i))^2$$

où $\hat{f}(x_i)$ est la réponse (la prédiction) du modèle (on note la prédiction $\hat{Y} = \hat{f}(X)$)

☞ La mesure MSE (donnée par Weka) est donc importante pour comparer 2 modèles (peut importe la méthode)

Illustrations (suite)

- Une de ses faiblesses : elle peut augmenter à cause de seulement quelques grosses erreurs !

- L'erreur MSE est (souvent) calculée et minimisée sur l'ensemble d'apprentissage (appelé *training_MSE*)

  ➛ mais il nous faut plutôt la MSE du test :

  ➛ c-à-d. la MSE sur des données qui n'ont pas servies à construire le modèle.

- Sur un un exemple non encore rencontré (x_0, y_0) :

$$test\_MSE = moyenne(y_0 - \hat{f}(x_0))^2$$

- Il n'y a pas de garantie que $test\_MSE \approx training\_MSE$

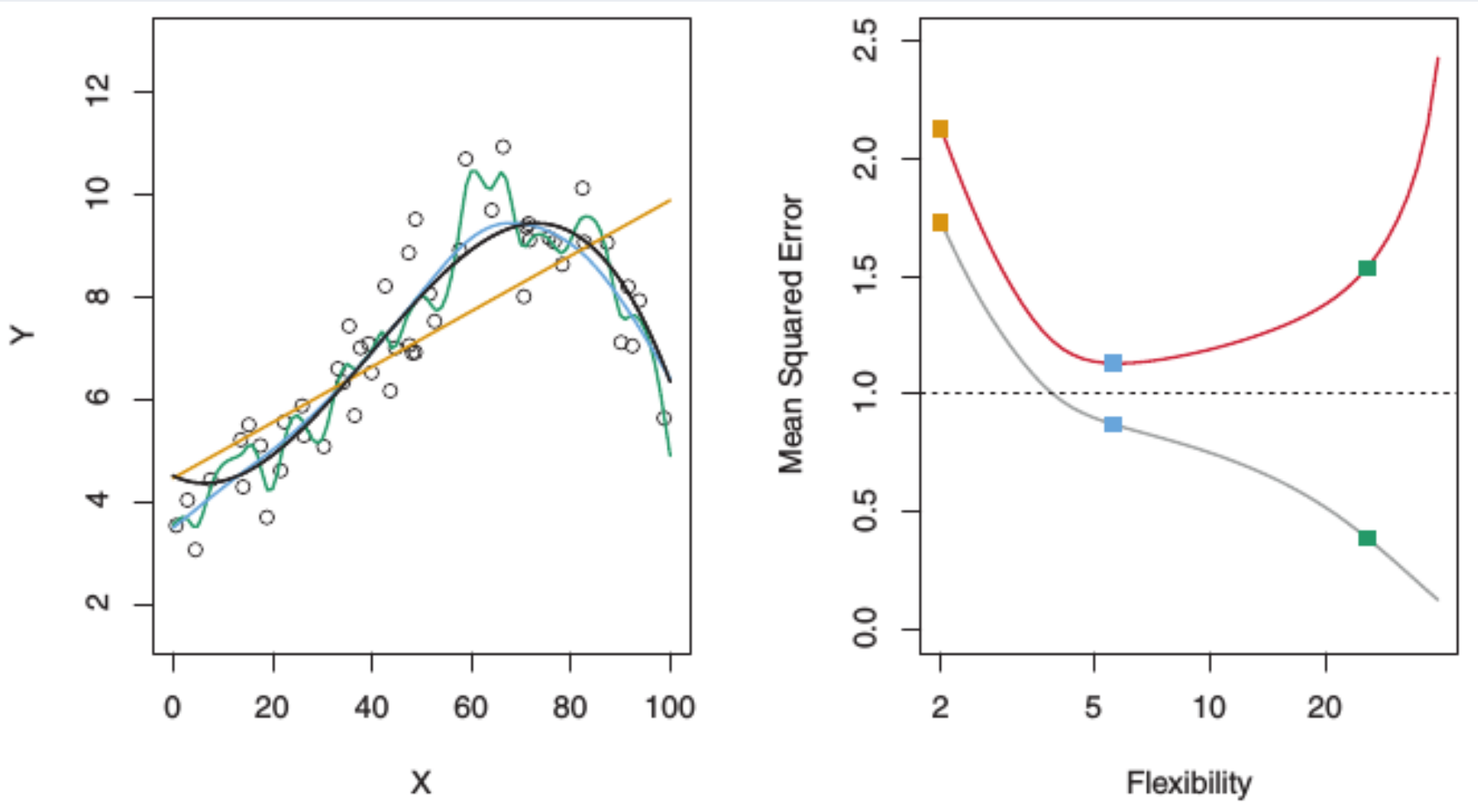
  ➛ Que faire si un ensemble de test n'est pas disponible ?

- Étude à travers 3 exemples ../..

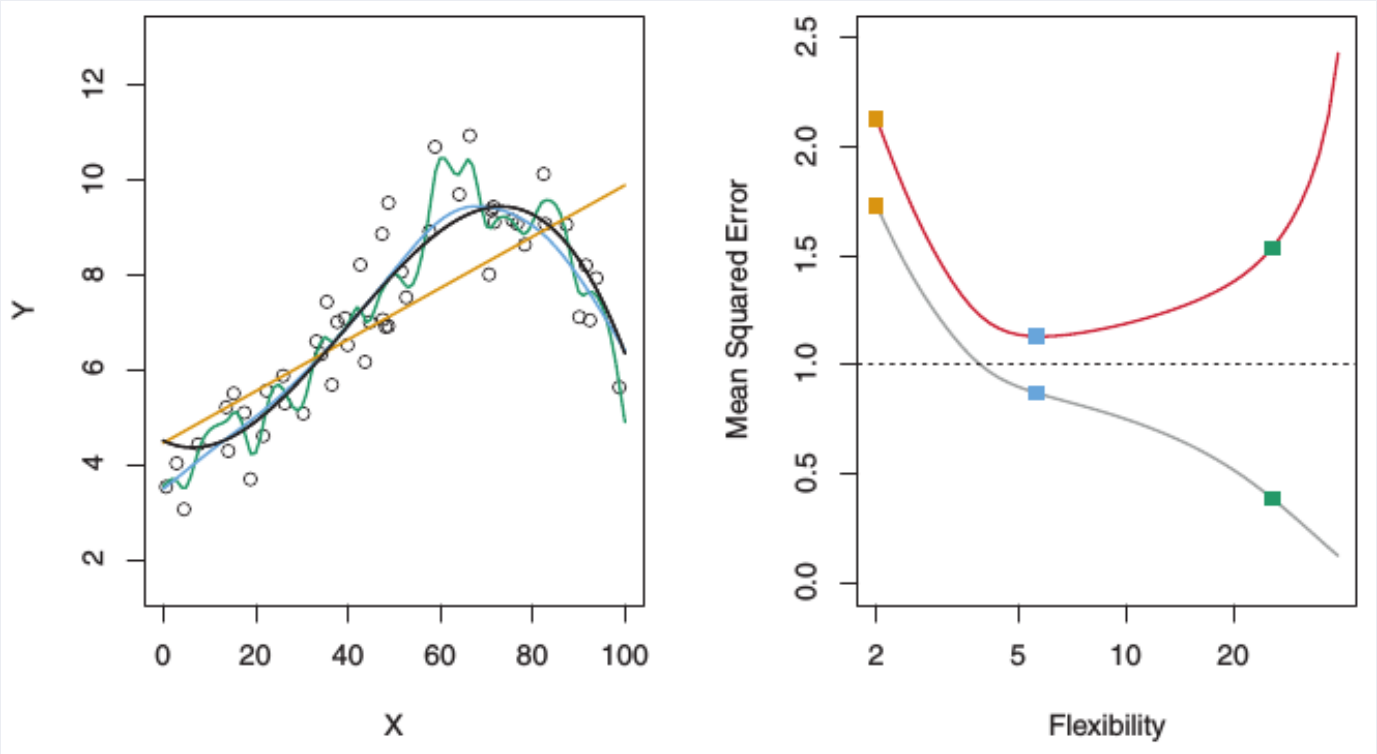
Illustrations (suite)

La courbe noire ci-dessus (à gauche) est une simulation de $y = f(x)$: $x =$ les ronds.

➟ On veut le modèle proche de cette courbe parmi les modèles appris (à gche).



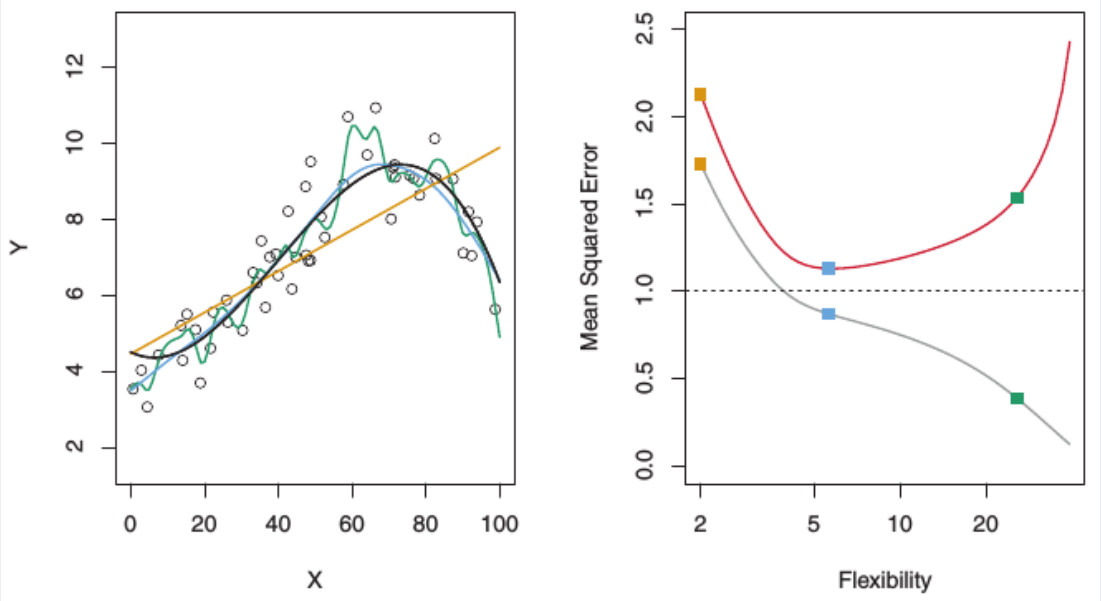
Illustrations (suite)



- A gauche : **les modèles appris**
  - la courbe **noire** = les données et la fonction $y = f(x)$ à apprendre ← **la référence**
  - 3 estimations de f : régression linéaire spline lissée et une autre spline
- A droite : **l'erreur en fonc. de la flexibilité**
  - En gris Train_MSE (en gris), en rouge Test_MSE de $\hat{f}$, et
  - En pointillée : le minimum (possible) de *Test_MSE* des 3 méthodes
  - Les carrées : *Train_MSE* et *Test_MSE* pour les mêmes courbes de couleur à gauche

Illustrations (suite)

• Les données (points ronds) obtenus par simulation de f (**connue**, en noir) avec sa ε (bruit, erreur, ...)



• La **flexibilité** d'une courbe $\Longleftrightarrow$ Son **degré de liberté**

• Pour augmenter la *flexibilité* de $\hat{f}$, il faut mesurer davantage de paramètres pour $\hat{f}$ (risque d'overfitting)

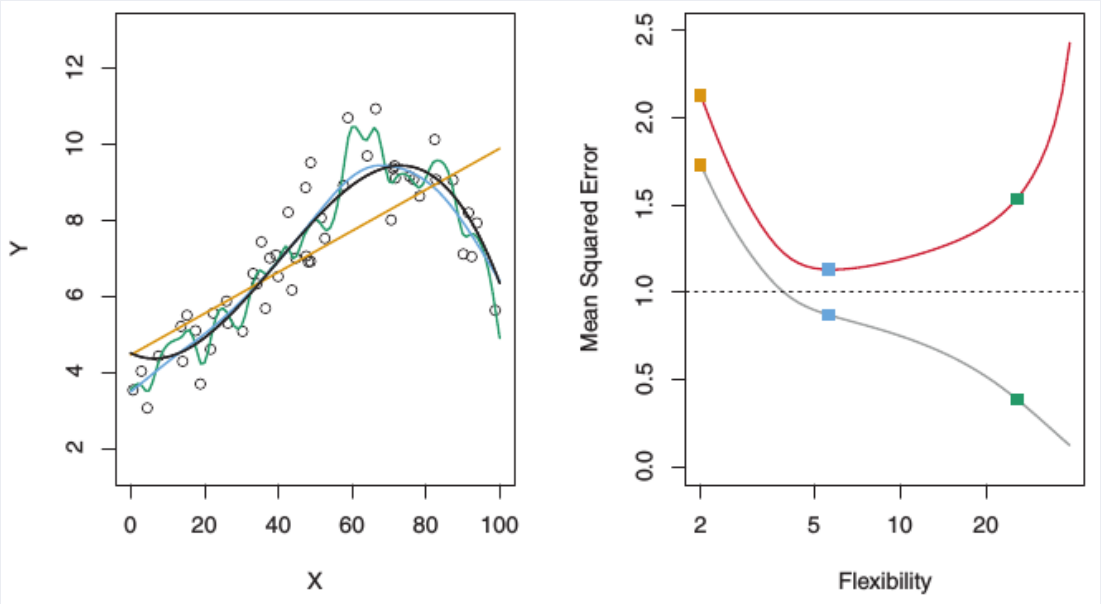
➛ Un tel modèle appris par coeur suit davantage les erreurs et les bruits dans les données

• Plus la flexibilité des splines augmente, mieux $\hat{f}$ colle aux données X (risque d'overfitting !)

• La courbe verte est davantage flexible mais s'éloigne de la vraie f (en **noir** à gauche).

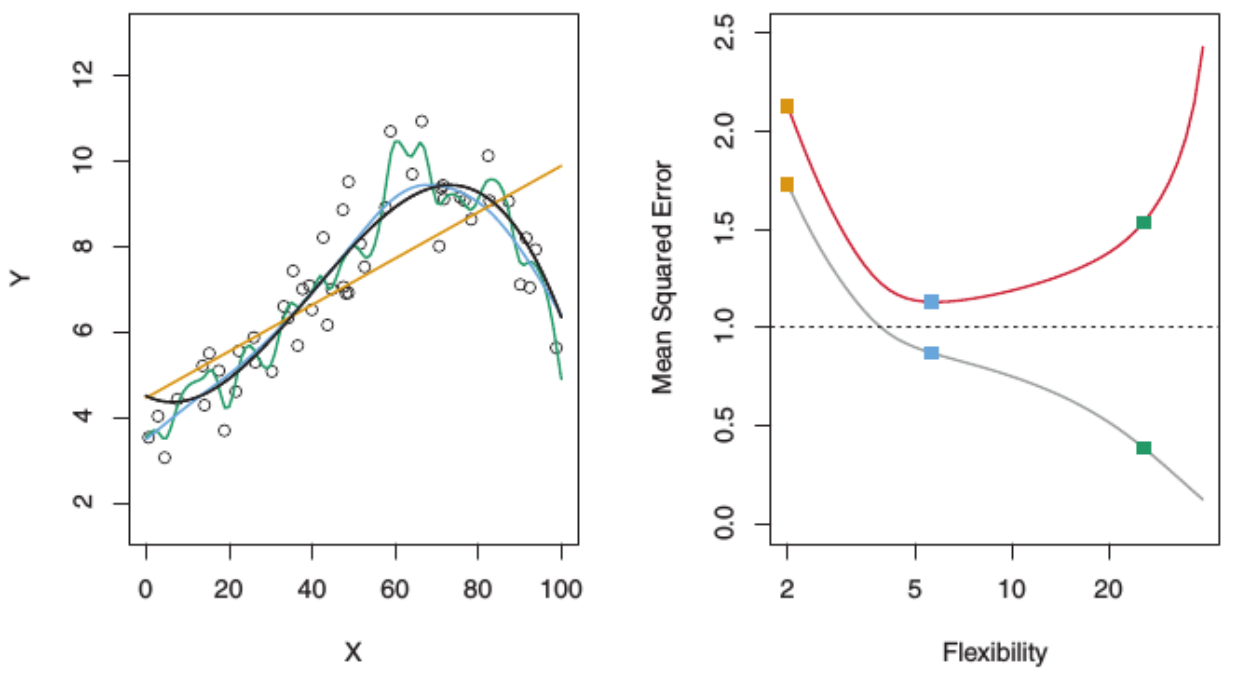
☞ En variant la flexibilité, on peut proposer de multiples $\hat{f}$

Illustrations (suite)



- A droite, la courbe grise est une fonction de la *flexibilité* (degré de liberté) pour nos splines
  ➛ La droite de régression linéaire a un minimum de degré de liberté (risque d'underfitting !)
- Le *Train_MSE* décline lorsque la flexibilité augmente (cf. courbe verte à gauche)
  ➛ La courbe verte a une petite erreur *Train_MSE* de toutes mais elle est la plus flexible.
- Dans cet ex., la courbe f (en noir) est connue : on peut mesurer *Test_MSE* (rouge à droite)
- Sur la *Test_MSE* (rouge à droite), le minimum de la *Test_MSE* est associée à la courbe bleu :
  ➛ Elle semble mieux estimer f
  ➛ La droite de régression et la courbe verte ont une grande *Test_MSE* (et une flexibilité opposée)

Illustrations (suite)



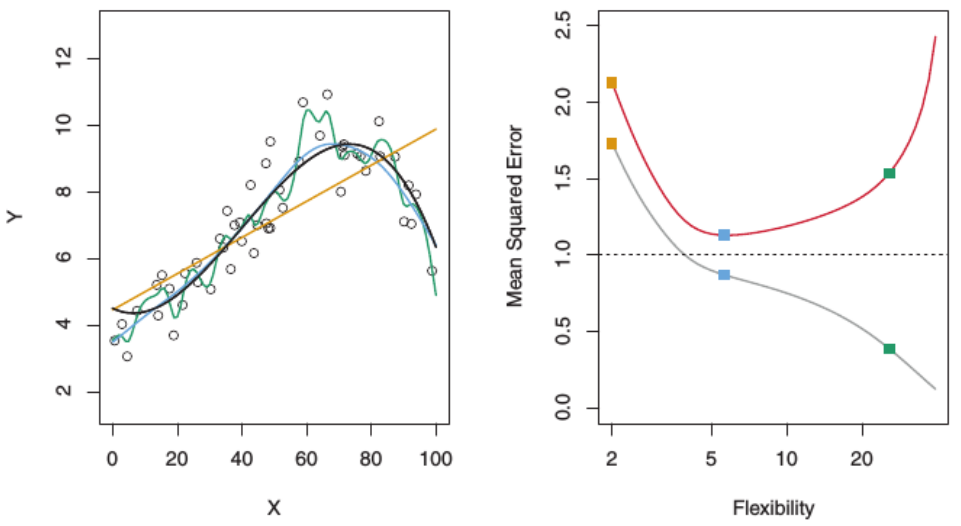
- L'horizontale pointillée (à droite) représente $Var(\varepsilon)$ irréductible

➛ = la plus petite *Test_MSE* pour tous les modèles proposés

➛ ε est utilisée dans la génération des points ronds à partir de f

- On constate que courbe bleu est une bonne estimation de f
- A droite, la flexibilité augmente avec la *Test_MSE*

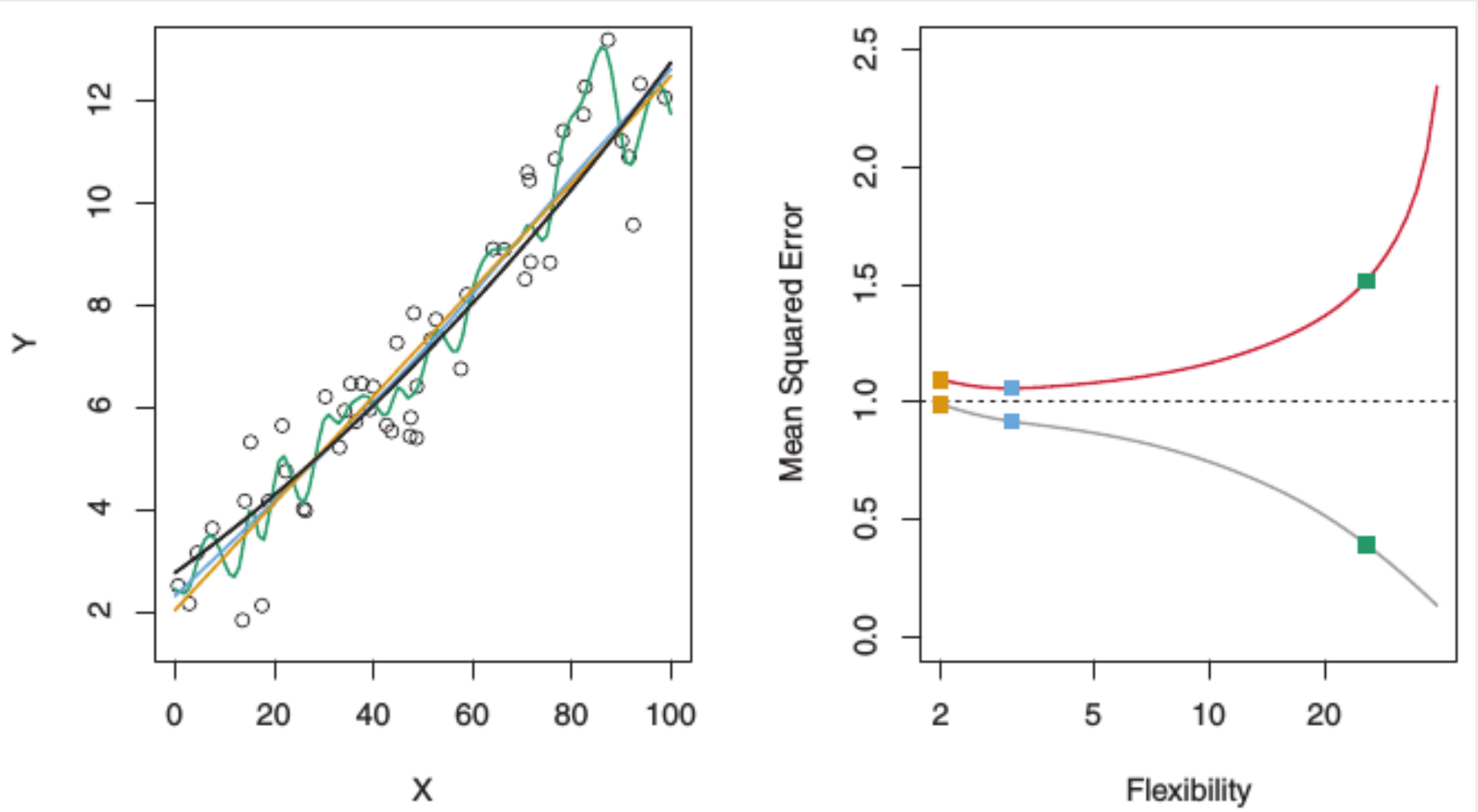
Illustrations (suite)



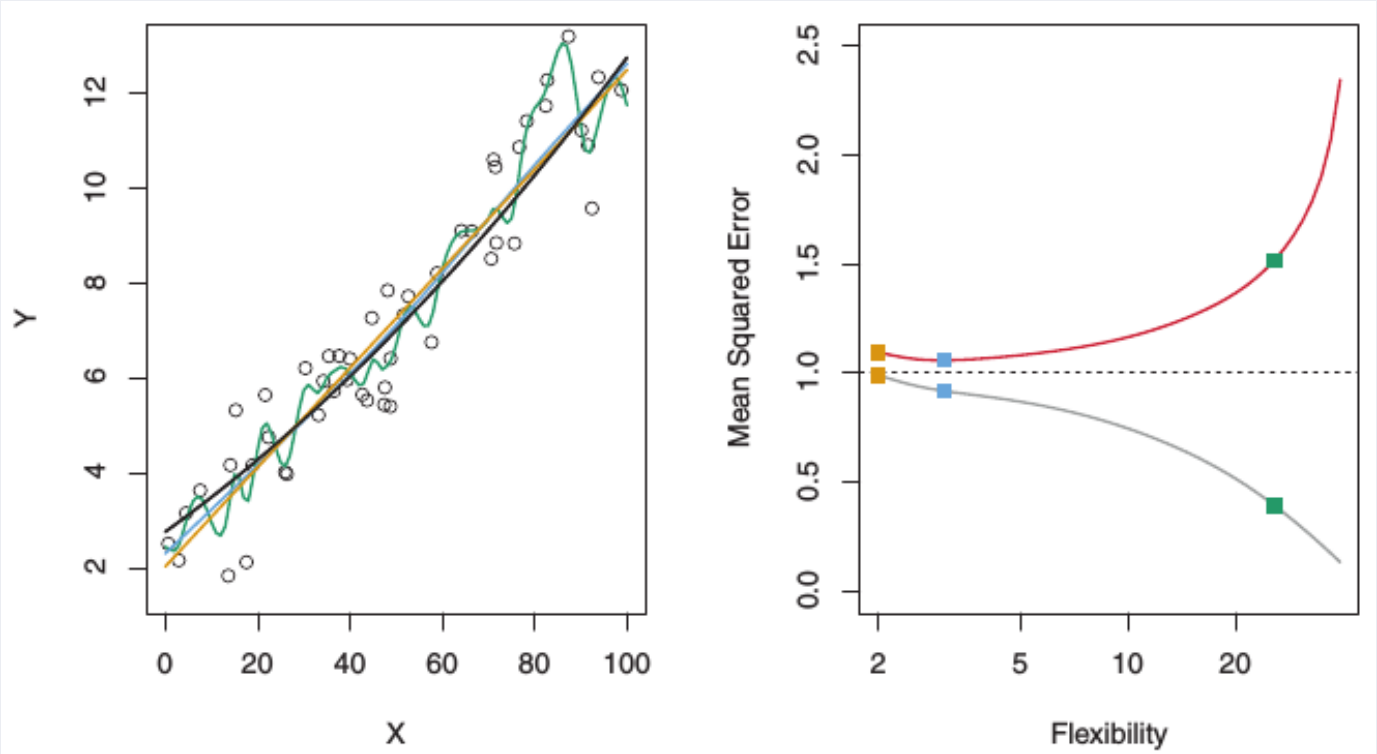
- Une **propriété fondamentale en apprentissage** qui est vraie quelque soit les données et la méthode utilisée.
  - ➛ Plus la flexibilité augment (on colle aux données), plus diminue la *Train_MSE*
  - ➛ Mais peut-être pas la *Test_MSE*.
- On a un *overfitting* si on a une petite *Train_MSE* mais une grande *Test_MSE*
  - ➛ Le modèle appris aura trop suivi le bruit qui ne se répète pas forcément dans les tests et les (vraies) données à venir
- En général, avec ou sans overfitting, *Train_MSE* est inférieure à *Test_MSE*
  - ➛ Car toute méthode cherche à minimiser cette erreur

Illustrations (suite)

2- Un autre exemple (avec le même code couleur)



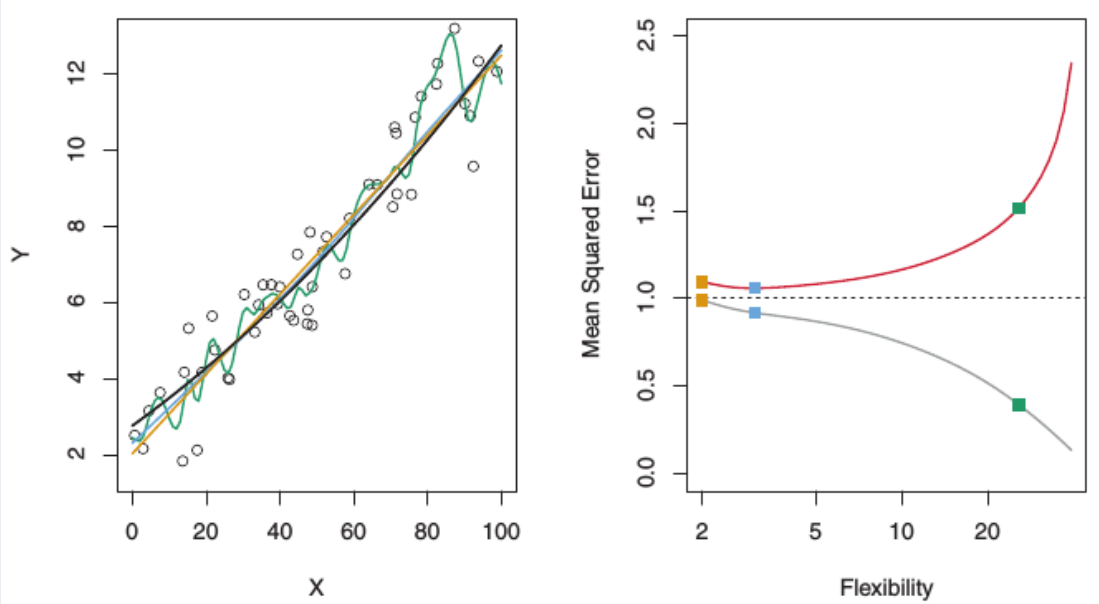
Illustrations (suite)



- A gauche :
  - la quasi droite **noire** = les données de la fonction $y = f(x)$ à apprendre
  - 3 estimations de f : régression linéaire spline lissée et une autre spline
- A droite :
  - En gris Train_MSE (en gris), en rouge Test_MSE de $\hat{f}$, et
  - En pointillée : le minimum général de *Test_MSE* des 3 méthodes
  - Les carrées : *Train_MSE* et *Test_MSE* pour les mêmes courbes de couleur à gauche

Illustrations (suite)

Comme pour l'exemple précédent : ici la "vraie" f est quasi linéaire



- La *Train_MSE* diminue avec monotonie quand la flexibilité augmente

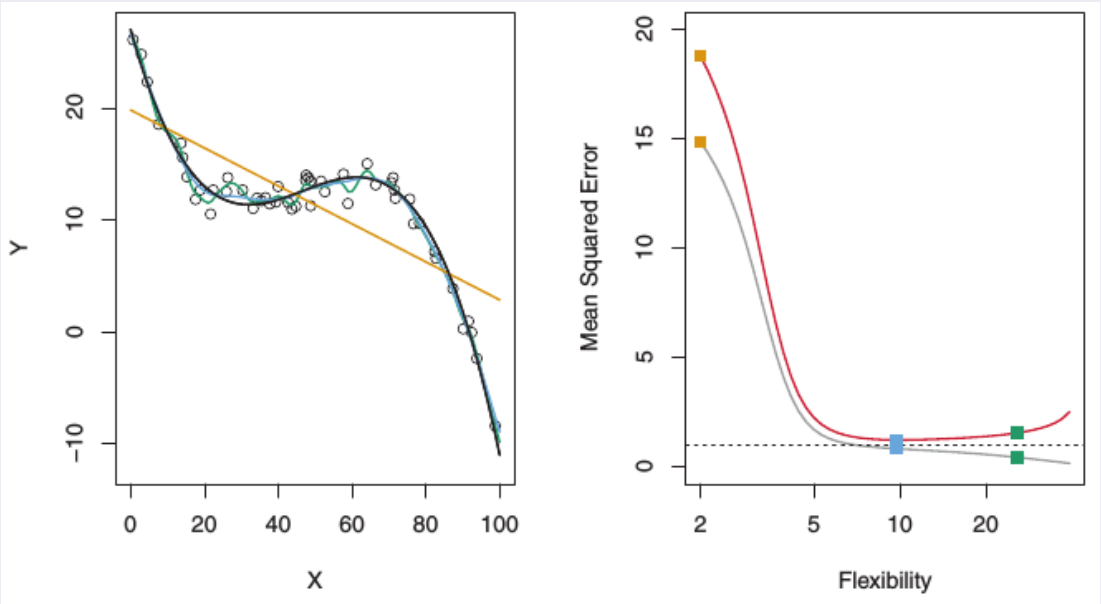
➛ De même que la courbe rouge en forme de U de *Test_MSE*

- Mais puisque la vraie f est quasi linéaire, la *Test_MSE* diminue lentement avant de d'augmenter

➛ L'estimation linéaire convient mieux que la courbe verte trop flexible

Illustrations (suite)

3-Un autre exemple (avec le même code couleur)



- Ici f (en noir à gauche) est non linéaire :
  ➛ Les *Train_MSE* et *Test_MSE* décroissent rapidement avant l'augmentation de *Test_MSE*

- Dans ces 3 exemples, la flexibilité correspondant au modèle avec un minimum *Test_MSE* peut varier selon la BD.
  ➛ Le but d'un apprentissage est de trouver ce point minimum.

Illustrations (suite)

• La courbe rouge en forme de U de *Test_MSE* est le résultat de deux propriétés statistiques des méthodes d'apprentissage.

• On peut montrer que $\mathbb{E}(Test\_MSE)$ pour un (nouveau) x_0 peut se décomposer en une somme de 3 quantités **fondamentales** :

la variance de $\hat{f}(x_0)$, la carrée du biais de $\hat{f}(x_0)$ et la variance de ε :

• On a :

$$\mathbb{E}(y_0 - \hat{y_0})^2 = \mathbb{E}[(f(x_0) + \varepsilon - \hat{f}(x_0))^2] = \mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] + Var(\varepsilon)$$

Exprimée sous la forme :

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = [Biais(f, \hat{f})]^2 + Var(\hat{f}(x_0)) + Var(\varepsilon)$$

• On obtiendrait cette *erreur moyenne attendue* si on testait (quasi) exhaustivement et itérativement une grande quantité de données de test.

• Pour minimiser $\mathbb{E}(Test\_MSE)$, on doit minimiser les deux premiers termes.
➛ Dans le meilleurs des cas, *Test_MSE* ne sera pas inférieure à $Var(\varepsilon)$

Illustrations (suite)

- Rappel : $\mathbb{E}(f - \hat{f})^2 = [Biais(f, \hat{f})]^2 + Var(\hat{f}) + Var(\varepsilon)$

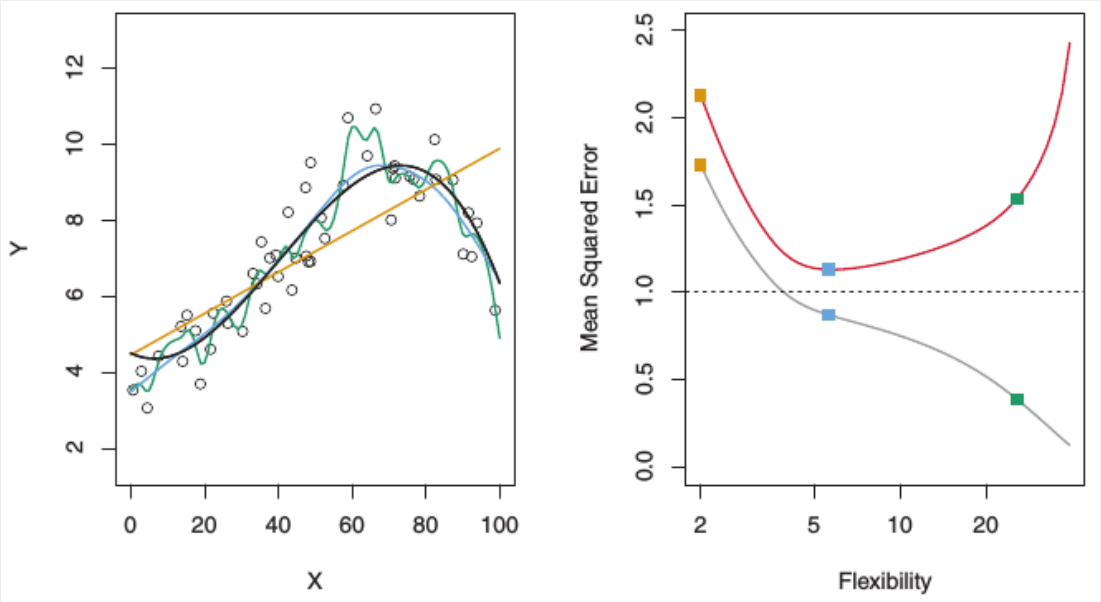
La variance représente les changement de $\hat{f}$ si on changeait d'ensemble de données.
- ➛ Généralement, si on change les données, on apprend une $\hat{f}$ différente
- ➛ Dans le cas idéal, $\hat{f}$ ne devrait pas trop changer, quand on change de BD.
  - ☞ Pour le même espace de données (e.g. tel type de maladie)

- En général, les méthodes les plus flexibles ont une variance élevée

- Si on considère les courbes bleue et verte du premier exemple ci-dessus :

◦ La courbe flexible verte a une variance élevée :
Elle colle trop aux données et si on en change une, $\hat{f}$ changera (grandement).

◦ Par contre, la droite linéaire est quasi inflexible avec une variance faible :
➛ Le changement d'une des données ne modifiera presque pas $\hat{f}$.



Illustrations (suite)

Le Biais :

• Rappel : $\mathbb{E}(f - \hat{f})^2 = [Biais(f, \hat{f})]^2 + Var(\hat{f}) + Var(\varepsilon)$

• Le Biais fait référence à l'erreur de $\hat{f}$ = approximation de données "réelles"

• Ces problèmes "réels" paraissent compliqués mais pourraient avoir un modèle simple.

• P. Ex. rarement un phénomène réel présente une simple relation entre ses paramètres par une régression linéaire

➛ Une $\hat{f}$ linéaire aura donc un Biais élevé dans l'estimation de f

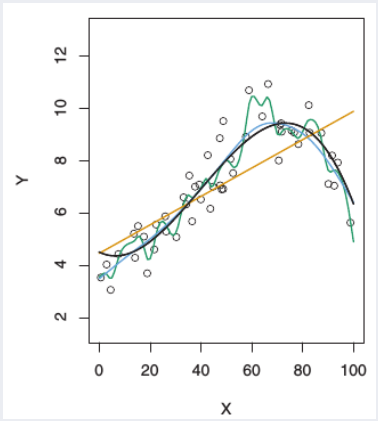
➛ Comme dans la 3e exemple ci-dessus, f est loin d'être linéaire.

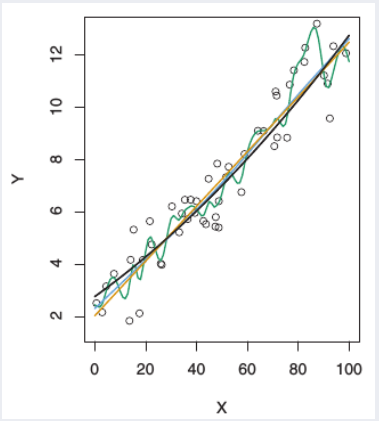
• Par contre, dans le 2e exemple, f est proche de la $\hat{f}$ linéaire.

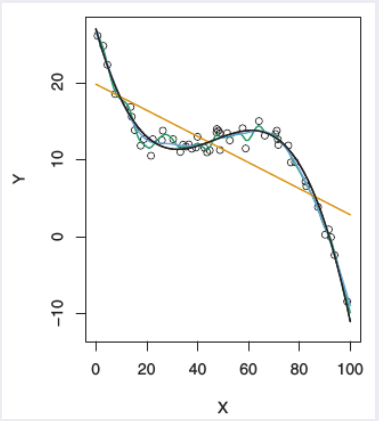
➛ Avec assez de données, $\hat{f}$ est capable de présenter f

➛ Les méthodes les plus flexibles (comme $\hat{f}$ linaire de cet exemple) a un Biais plus petit.

Illustrations (suite)



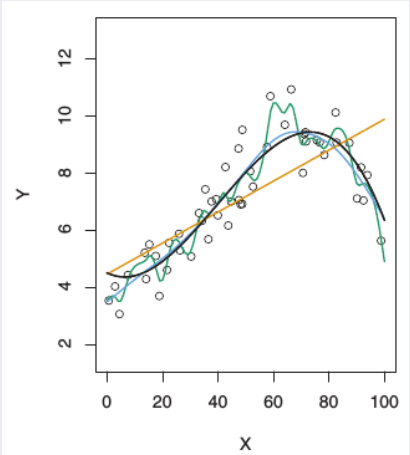


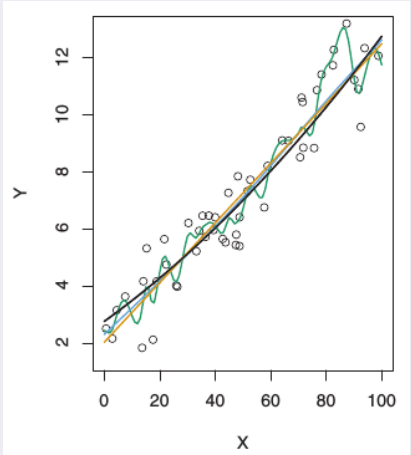


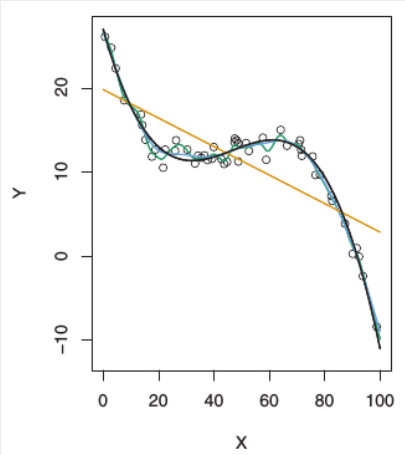
- Plus on utilise une méthode flexible (avec peu de paramètres), plus on a une variance élevée et un Biais plus petit.

- Le rapport entre les deux détermine les variations de *Test_MSE*.
  - Si la flexibilité augmente, le Biais commence à décroître plus rapidement que la variance augmente.
  - Et le *Test_MSE* diminue
  - Mais à un certain point, l'augmentation de la flexibilité a un petit effet sur le Biais mais la variance commence à à augmenter.
  - Dans ce cas, *Test_MSE* augmente
  - On a observé ce phénomène de baisse puis augmentation de *Test_MSE* dans les 3 exemples.

Illustrations (suite)



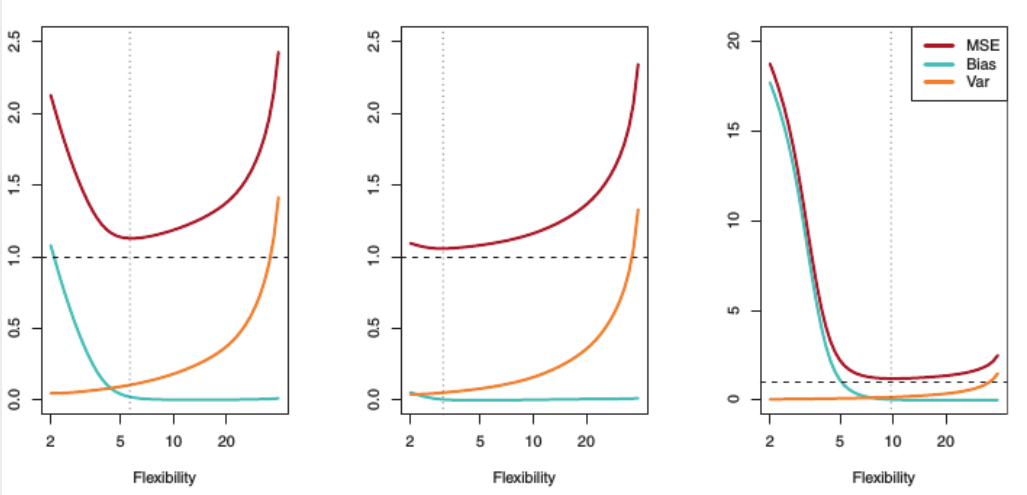




- La triple courbe ci-dessous illustre, pour ces 3 exemples rappelés ci-dessus, l'équation

$$\mathbb{E}(f - \hat{f})^2 = [Biais(f, \hat{f})]^2 + Var(\hat{f}) + Var(\varepsilon)$$

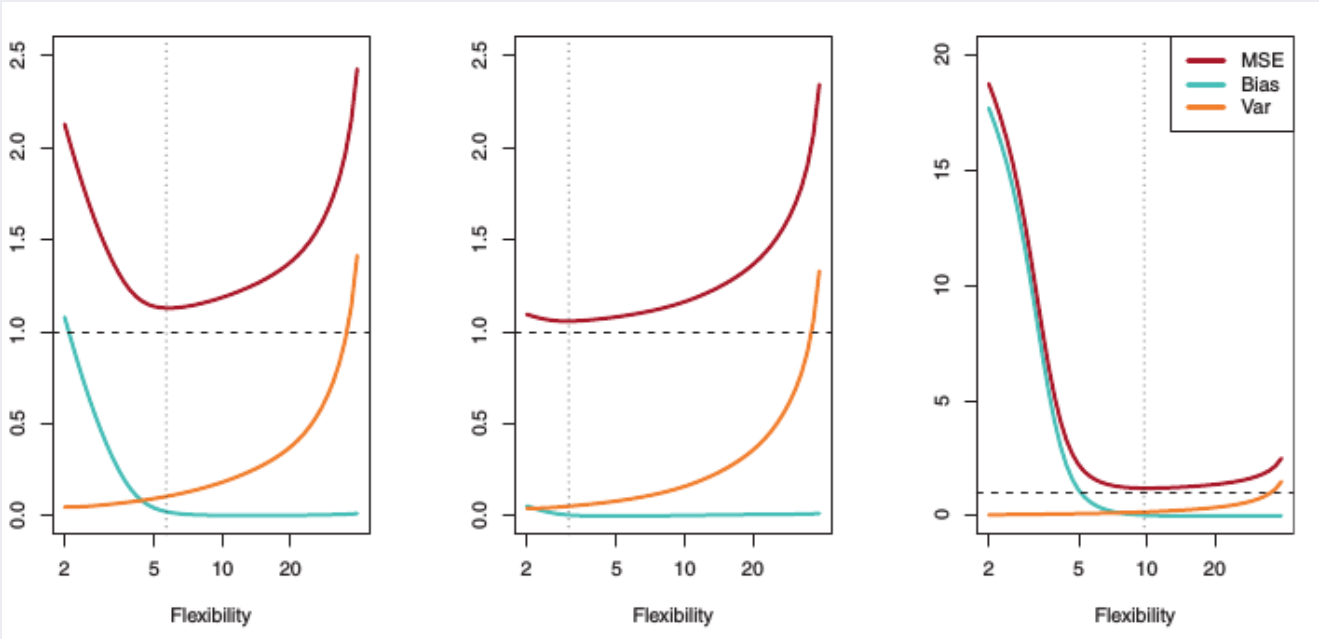
$Biais^2$ en bleu, Var en darkorange



La pointillée verticale = la flexibilité correspondant à la plus petite $Test\_MSE = Var(\varepsilon)$ sur la courbe rouge

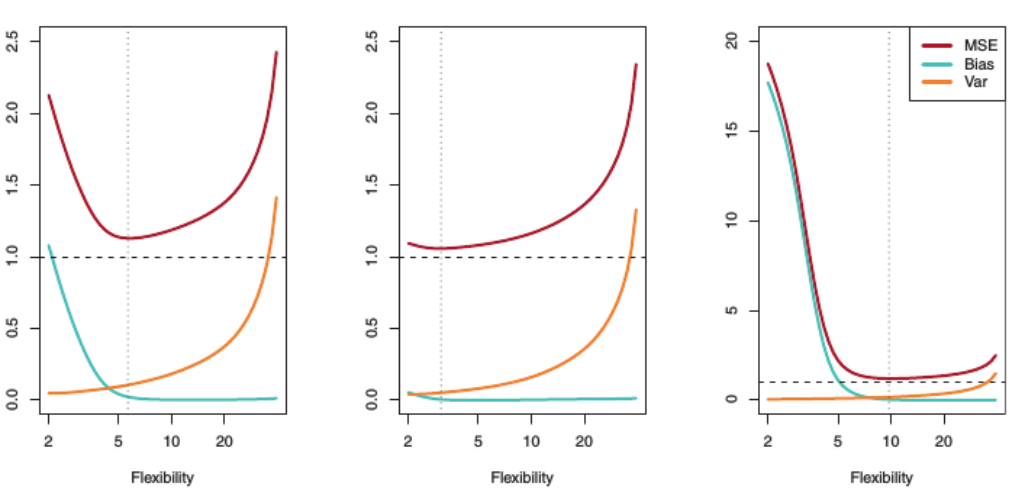
Illustrations (suite)

Pour les 3 exemples :



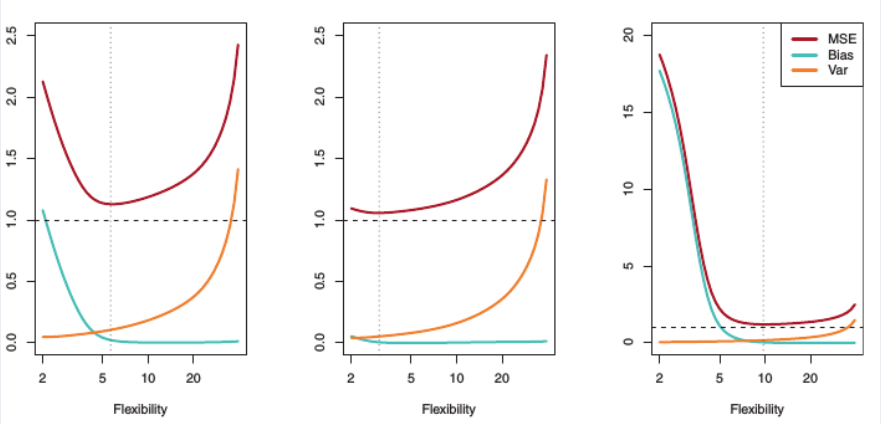
- Ici, la courbe rouge représente l'erreur $\mathbb{E}(f - \hat{f})^2$
- Dans les 3 figures, quand la flexibilité augmente, la variance augmente et le Biais diminue.
- La flexibilité correspondant à *Test_MSE* optimale est très différente pour les 3 cas.
  ➛ Car $Biais^2$ et la variance changent pour ces 3 différentes BDs.

Illustrations (suite)



- <u>A gauche</u> : le Biais diminue faisant baisser *Test_MSE*
- <u>Au centre</u> : f est proche de linéaire ;
  il y a une petite baisse de $Biais^2$ lorsque la flexibilité augmente et *Test_MSE* diminue
  très peu avant d'augmenter rapidement comme pour la variance
- <u>A droite</u> : quand la flexibilité augmente, $Biais^2$ diminue fortement car le vraie f n'est pas linéaire.
  De même, la variance et la flexibilité augmentent peu.

Illustrations (suite)



- Le **compromis Biais-Variance** cherche à minimiser les deux
  - On peut trouver un Biais faible et une variance élevé en **overfitting**
  - On peut trouver un Biais élevé et une variance faible en **underfitting**
- On **ne peut pas** facilement trouver ces deux mesures dans les cas réels Mais
  - Si l'exploration des données montre une comportement quasi linéaire, on choisit une méthode peu flexible
  - Si la BD est non-linéaire, préférez les méthodes moins flexible (avec plus de paramètres)
- Dans tous les cas, la X-V est une bonne méthode d'estimation de *Test_MSE*

Illustrations (suite)

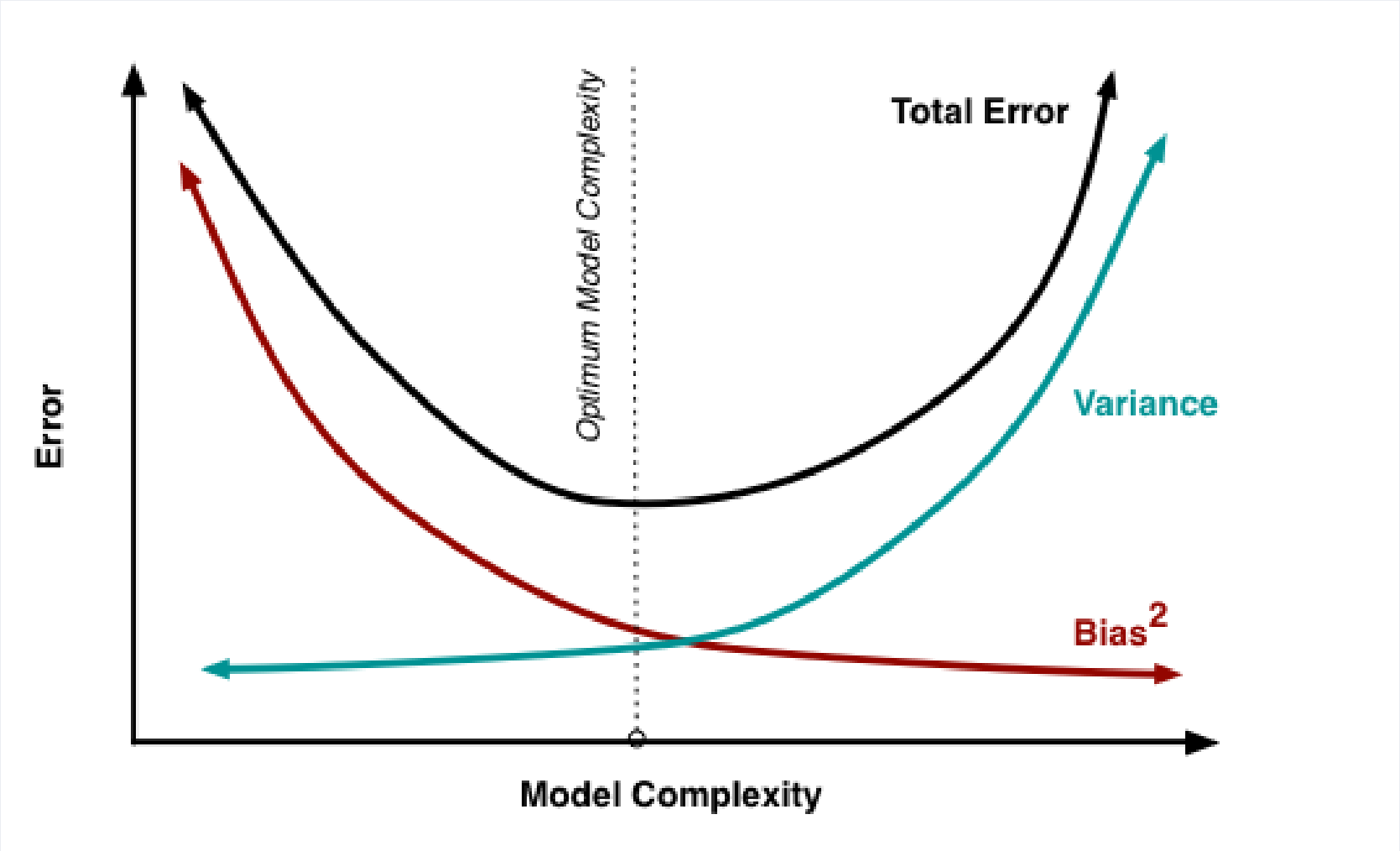


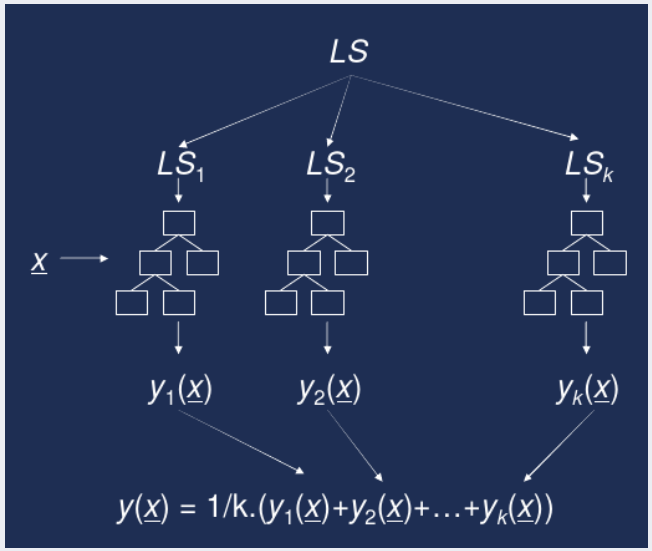
FIGURE 1: Contribution du couple Biais / Variance à l'erreur du modèle (complexité = Flexibilité)

Réduction de la variance

Idées générales :

1. Sur un même-modèle, réduire le risque d'overfitting par
   - Élagage
   - Arrêt prématuré contrôlé (seuil de profondeur d'arbres, nbr. de règles, ...)
   - Régularisation (en réduisant l'espace d'hypothèses)
   - X-validation
2. En multi-modèles (Bagging : Bootstrap AGGregatING)
   - L'idée : le modèle moyen ($\mathbb{E}_{LS}\{y\}$) a la même erreur ($biais^2$) mais sa variance tend vers 0.
   - L'idéal : il faut avoir une infinité d'ensembles d'apprentissage LS mais souvent, on a un seul ensemble :
     ➞ On simule le cas idéal par un échantillonnage
     ➞ *Bootstrap sampling* : échantillonnage avec remise de n instances de la B.D. de taille n
   - Peut mener à une perte de propriétés du modèle unique
   - Illustration : ../..

Réduction de la variance (suite)



- Quelques exemples pratiques :
  - ★ Bagging (ci-dessus)
  - ★ Random Trees (choix aléatoire dans l'ensemble de test)
  - ★ Random Forest (échantillonnage d'attributs)
  - ★ Initialisation Random des pondérations dans les RNs
  - ★ Etc

Addendum : calculs Biais-Variance

Détails du calcul du couple Biais-Variance :

Et si on ré-écrit (ajouter et enlever : termes en bleu) :

$$\begin{aligned}
\mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] &= \mathbb{E}\left[\left(f(x_0) - \mathbb{E}(\hat{f}(x_0)) + \mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)^2\right] \\
&= \mathbb{E}\left[\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right) + \left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)^2\right] \\
&= \mathbb{E}\left[\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right)^2 + 2.\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right).\left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)\right. \\
&\qquad \left. + \left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)^2\right] \\
&= \mathbb{E}\left[\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right)^2\right] + 2.\mathbb{E}\left[\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right).\left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)\right] \\
&\qquad + \mathbb{E}\left[\left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)^2\right]
\end{aligned}$$

Sachant que $f(x_0) - \mathbb{E}(\hat{f}(x_0))$ est constante, on aura (pour le terme au milieu) :

$$\begin{aligned}
&2.\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right).\mathbb{E}\left[\left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)\right] && \text{On sort la cste.} \\
&= 2.\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right).\left[\left(\mathbb{E}(\hat{f}(x_0)) - \mathbb{E}(\hat{f}(x_0)\right)\right] = 0 && \text{On rentre } \mathbb{E}
\end{aligned}$$

L'expression de l'erreur devient :

$$\mathbb{E}[(f(x_0) - \hat{f}(x_0))^2] = \mathbb{E}\left[\left(f(x_0) - \mathbb{E}(\hat{f}(x_0))\right)^2\right] + \mathbb{E}\left[\left(\mathbb{E}(\hat{f}(x_0)) - \hat{f}(x_0)\right)^2\right]$$

Et la totalité de l'erreur (avec $Var(\varepsilon)$) est exprimée par

$$\mathbb{E}(y_0 - \hat{f}(x_0))^2 = [Biais(f, \hat{f})]^2 + Var(\hat{f}(x_0)) + Var(\varepsilon)$$

Addendum : calculs Biais-Variance (suite)

D'où viennent ces résultats ?

- Sachant que : $Var[Z] = \mathbb{E}[Z^2] - \mathbb{E}[Z]^2$ l'expression alternative de la variance $\forall Z$

  ➛ On en obtient $\mathbb{E}[Z^2] = Var[Z] + \mathbb{E}[Z]^2$

- On a également :
  - $\mathbb{E}[f] = f$ (pour f supposée déterministe dans la cas d'une BD donnée = i.e une cste.)
  - $\mathbb{E}[\varepsilon] = 0$ (vu plus haut : moyenne nulle de ε)
  - $\mathbb{E}[Y] = \mathbb{E}[f(X) + \varepsilon] = \mathbb{E}[f(X)] + \mathbb{E}[\varepsilon] = f(X) + 0 = f(X)$ (cf. $Y = f(X) + \varepsilon$)

- Également, par la formulation habituelle de la variance, on a :

$$Var[Y] = \mathbb{E}[(Y - \mathbb{E}[Y])]^2 = \mathbb{E}[(Y - f(X))^2] = \mathbb{E}[(f(X) + \varepsilon - f(X))^2]$$
$$= \mathbb{E}[\varepsilon^2] = Var[\varepsilon] + \mathbb{E}[\varepsilon]^2 = Var[\varepsilon] = \sigma^2$$

(cf. ci-dessus : $\mathbb{E}[\varepsilon] = 0$ ☞ $\mathbb{E}[\varepsilon^2] \neq \mathbb{E}[\varepsilon]^2$)

- Expression du couple Biais-Variance (calculée différemment) :

$$\mathbb{E}[(f - \hat{f})^2] = \mathbb{E}[f^2 + \hat{f}^2 - 2f\hat{f}]$$
$$= \mathbb{E}[f^2] + E[\hat{f}^2] - \mathbb{E}[2f\hat{f}]$$
$$= Var[f] + \mathbb{E}[f]^2 + Var[\hat{f}] + \mathbb{E}[\hat{f}]^2 - 2f\mathbb{E}[\hat{f}]$$ (et si $\mathbb{E}[f] = f$, v. ci-dessus)
$$= Var[f] + Var[\hat{f}] + (f - \mathbb{E}[\hat{f}])^2$$ (on a $\mathbb{E}[f] = f$ d'où $f - \mathbb{E}[\hat{f}] = \mathbb{E}[f] - \mathbb{E}[\hat{f}]$)
$$= Var[f] + Var[\hat{f}] + \mathbb{E}[f - \hat{f}]^2$$
$$= Var(\varepsilon) + Var[\hat{f}] + Bias[\hat{f}]^2$$

Calcul du coût

- Jusqu'ici : évaluation sans tenir compte du coût.
- Ici :
  - Coût de mauvaise décision ou de mauvaise classification
  - Ce n'est pas la même chose que la probabilité de la prédiction.
- Ne pas tenir compte des vrais coûts peut conduire à certains résultats étranges.
- **Exemples :**
  - Décision de prêt : le coût de prêter à un client "défaillant" est bien plus grand que le coût de ne pas prêter à un bon client.
  - Détection de tâche de pétrole en mer : le coût d'échec dans la détection est bien plus important que le coût d'une fausse alerte.
  - Prévision de charge : le coût de production d'une quantité plus importante que nécessaire est bien MOINS important que de ne pas préparer une forte demande.
  - Diagnostic de pannes : le coût d'une mauvaise identification est moindre que celui de ne pas détecter une machine qui va tomber en panne.

Calcul du coût (suite)

D'autres exemples :

◦ Mails promotionnels : le coût d'envoi de mails sauvages (junk) à un foyer qui ne répond pas est bien moindre que le coût de ne pas envoyer à un foyer qui aurait répondu

➙ Si les coûts sont connus, on peut / il faut en tenir compte

◦ Ex. des Vaches : déterminer le "jour" exact où chaque vache est en rut

Attributs : volume de lait/composition chimique/ordre de arrivée en traite

Indication : les vaches présentées tjs. dans le même ordre sauf en cas de "rut" ...

Une règle simpliste : une vache n'est jamais en rut (97% correcte) !

➙ Mais il est important de prédire exactement quand une vache est en rut

☞ Les 2 coûts (lait/rut) différents dans la réalité
mais la classification suppose *le même coût d'erreur* pour les 2.

➙ Il vaut mieux tenir compte de différents coûts

Matrices de confusion / coûts

Rappels matrice de confusion

- Cas bi-classes : "oui/non", "prêter/pas", ...
  - Les 4 différents cas possibles d'une seule prédiction → matrice de confusion.
  - Le "vrai positif" (TP) et "vrai négatif" (TN) sont les classifications correctes.
  - "faux positif" (FP) et "faux négatif" (FN) = mauvaises classifications.

Classes Prédites		*Oui*	*Non*
Vraies classes	*Oui*	Vrai Positif (**TP**)	Faux Négatif (**FN**)
	Non	Faux Positif (**FP**)	Vrai Négatif (**TN**)

TABLE 1: Matrice de confusion pour deux classes

  - Dans le cas de multi-classes : des lignes et colonnes pour **chaque classe**.
  - Les bons résultats correspondent à la somme le long de la diagonale et les mauvais (nombres petits, idéalement zéros) à celle le long de la diagonale inverse.

Matrices de confusion / coûts (suite)

• Les 2 types d'erreur **fausse positive** et **fausse négative** ont en général des **coûts différents** (comme les 2 types de classifications correctes TP et TN avec des bénéfices différents).

• **Taux de succès** = la somme des **TP** et des **TN** divisée par le total des prédictions de l'ensemble de test : $\frac{TP+TN}{TP+TN+FP+FN}$

• Le calcul des coûts permet de connaître la moyenne des coûts par décision.

• Y ajouter le coût de l'Apprentissage Automatique + coût d'assemblage / intégration des données (lorsque tous les coûts sont connus)

Contenu d'une matrice des coûts :

$C(i\|j)$		*Classe=Oui*	*Classe=Non*
Vraies classes	*Oui*	$C(Oui\|Oui)$ (TP)	$C(Oui\|Oui)$ (FN)
	Non	$C(Oui\|Non)$ (FP)	$C(Non\|Non)$ (TN)

TABLE 2: Matrice de coûts pour 2 classes : $C(i|j)$ = coût de (mal) classer en I un ex. de J

Matrices de confusion / coûts (suite)

Deux exemples concrets :

$C(i\|j)$		$classe = +$	$classe = -$
Vraies classes	$+$	-1	100 (pour FN)
	-	1	0

TABLE 3: Un exemple de matrice de coûts pour 2 classes

Avec coûts inversement proportionnels aux taux de succès :

	$C(i\|j)$	$+$	-
Vraies classes	$+$	150	40
	-	60	250

TABLE 4: Succès = 80%, Coût = 3910

	$C(i\|j)$	$+$	-
Vraies classes	$+$	250	45
	-	5	200

Succès = 90%, Coût = 4255

Coût vs. taux de succès

- Association de coûts pour la matrice de confusion

Compte		*Oui*	*Non*
Vraies classes	*Oui*	a	b
	Non	c	d

Coûts		*Oui*	*Non*
Vraies classes	*Oui*	p	q
	Non	q	p

- Le taux de succès est proportionnel au coût si

$$C(Oui|Non) = C(Non|Oui) = q \qquad \text{et}$$
$$C(Oui|Oui) = C(Non|Non) = p$$

➙ $N = a + b + c + d$, Taux de Succès $= (a + d)/N$

➙ Coût $= p(a + d) + q(b + c) = p(a + d) + q(N - a - d)$

$$= q.N - (q - p)(a + d)$$

$$\boxed{= N[q - (q - p) \times succes]}$$

Diagramme Lift (Lift Chart)

Un calcul orienté-coût

Exemple : résultats d'envoi de courriel promotionnel (mailing) à 1 000 000 foyers dont la plupart ne répondront pas (sous ensembles identifiés par un outil DM) :

- Cas 1 : seul 0,1% répondent = 1000 réponses (cas général si *mass-mailing*)
- Cas 2 : seul 0,4% de 100000 foyers prometteurs répondent = 400 réponses.

- Un diagramme **Lift Chart** permet de comparer (visuel) ces résultats.
  - Pour le cas 2, on dit (en Marketing) : on a un **facteur 4** d'augmentation de réponse
  - Appelé *le facteur de* ***lift*** du schéma d'apprentissage .
- Cas 3 : seul 0,2% de 400000 (premiers) répondent = 800 réponses (facteur de 2)

- Savoir si l'on doit envoyer une pub à ces foyers dépend des coûts.
  ➙ Il faudra comparer le coût du mailing avec les bénéfices attendues/engendrées

- Parfois : si l'on tient compte de tous les coûts ... :
  ➙ Le *mass mailing* peut être plus intéressant !

Génération du Lift Chart

But : Étant donné un schéma de DM qui calcule (des probabilités pour) les classes prédites de chaque instance (e.g. Bayésien Naïf, Régression Logistique, etc) :

- **Désigner des sous ensembles** qui ont un grand nombre d'instances positives,
- Idéal : nombre plus grand que le taux général sur tout l'ensemble de test .

Méthode : ordonner les instances selon leur probabilité (prédite) d'être **TP**

Ex : 100 instances, taux de succès = 20% (le non de la 3e ligne est un FP)

No	Probabilité Prédite	Vraie classe	No	Probabilité Prédite	Vraie classe
1	0.95	oui	...	...	...
2	0.93	oui	9	0.80	non
3	0.93	non	10	0.79	oui
4	0.88	oui	11	0.77	non
...	...	...	...	...	...

TABLE 5: Liste de données pour Lift Chart (ordonnée sur la probabilité)

➛ Probabilité : on peut penser à un *score* calculé pour chaque instance (pour les 2 classes).

➛ Weka peut les donner

Génération du Lift Chart (suite)

- On prend le sous ensemble d'une taille donnée avec la plus grande proportion possible d'instances positives ("Oui") en commençant depuis le début de la liste (les plus fortes probas en premier).
- Si la classe de chaque instance est connue, on a :

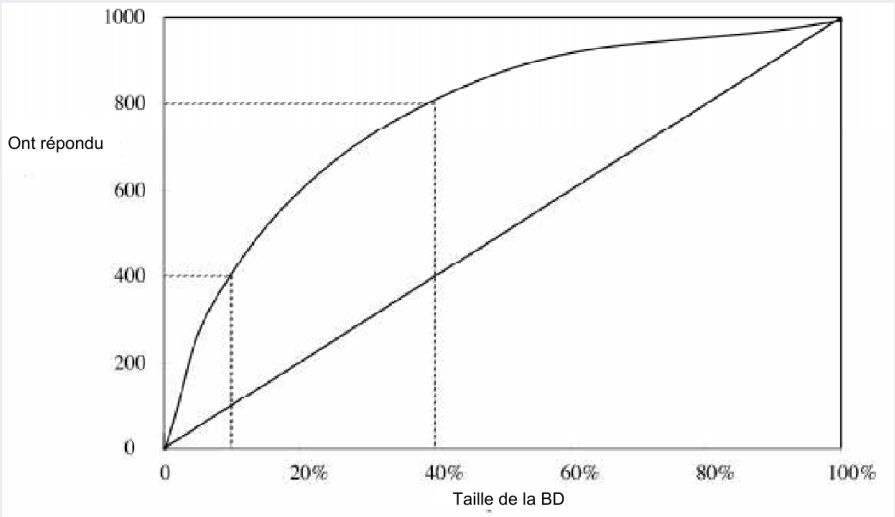
$$\textbf{Taux de succès} = \frac{\textit{nombre d'instances positives du sous ensemble}}{\textit{taille de tout l'ensemble de test}}$$

$$\textbf{Facteur de Lift} = \frac{\textit{Taux de succès}}{\textit{Taux de succès global (pour tout l'ensemble de test)}}$$

- Par exemple, on prend les 10 premiers "Oui" (dont 2 "non")
  - ➙ Taux de succès $= \frac{8}{10} = 80\%$,
  - ➙ Facteur de Lift $= \frac{80\%}{20\%} = 4$ (4 fois le taux 20%)
- Si on connaît tous les coûts :
  - ➙ Choisir le meilleur sous-ensemble (de taille/taux/facteur différents)
- L'idée : on cherche à réduire les coûts en ciblant les meilleurs sous-ensembles.

Génération du Lift Chart (suite)

• Pour avoir le diagramme *Lift Chart* (ci-contre), faire ces calculs pour divers sous-ensembles .
  ◦ L'axe horizontal est la taille (en % du total *mailout* possible).
  ◦ L'axe vertical : le nombre de réponses obtenues.
  ◦ La diagonale : les réponses théorique attendues (20% de succès)
  ◦ Meilleur coin : coin supérieur gauche = 100% succès



• Le coin bas gauche : "pas de mail du tout".

• Le coin haut droit : mailing total avec 1000 réponses (e.g. sur 5000 mailout, 20%).

• La diagonale = les réponses attendues (selon la taille des sous-ensembles aléatoires avec un taux de succès général de 20%).

• Permet de ne pas choisir des sous-ensembles aléatoires mais, plutôt :
  ◦ choisir des instances susceptibles de donner des réponses positives
  ◦ la courbe correspond au cumul des vraies réponses (la liste ordonnée des probas).

Génération du Lift Chart (suite)

Rappel de l'exemple :

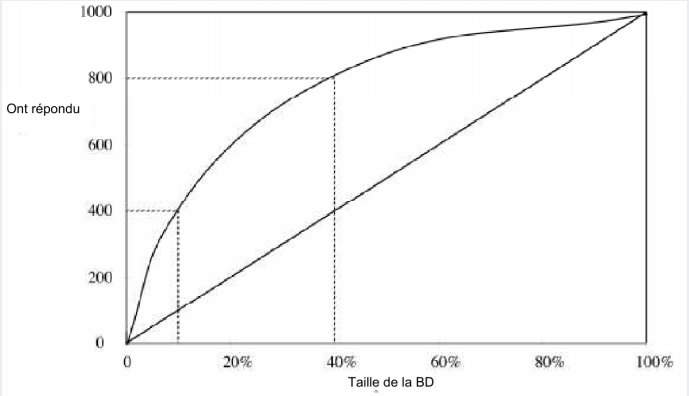


FIGURE 2: Un Lift Chart (avec les 2 points discutés : 400 réponses pour 10%, 800 réponses pour 40%)

- **La partie opérationnelle est le triangle supérieur.**
- L'axe X est la taille de l'échantillon (mailout), et Y le nombre des TP.
- **Meilleur coin possible (théorique)** : près du coin supérieur gauche = 100% succès

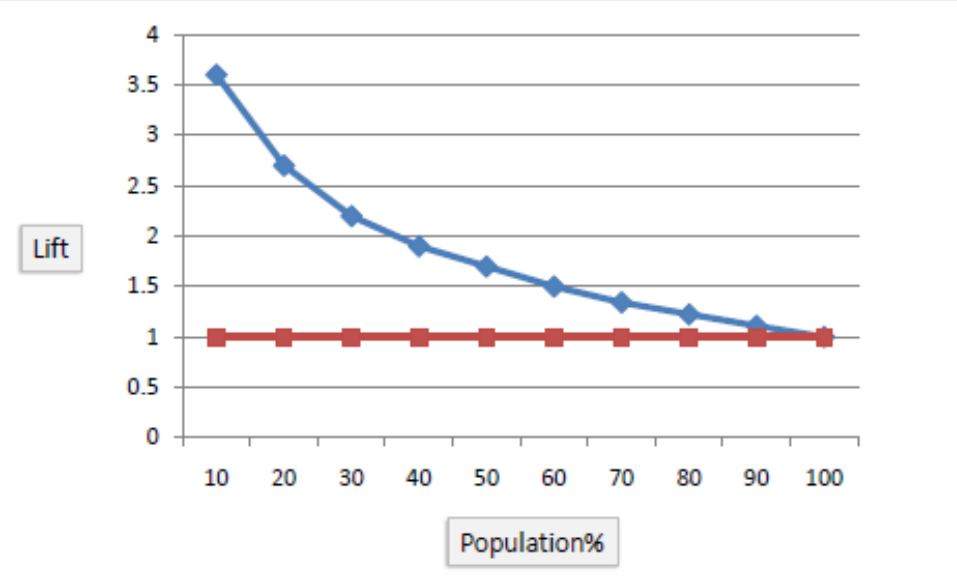
  ➙ Idéal. : 1000 réponses pour 1000 mails (20% des 5000) → qui répondent !
- Un bon outil DM doit nous placer au dessus de la diagonale,

  ➙ sinon, on aura une réponse moins bien que l'échantillonnage aléatoire.

Génération du Lift Chart (suite)

On peut donner une présentation synthétique du facteur de lift par un autre graphique :

- Exemple 2 : dans l'exemple suivant, le modèle prédictif a permis de cibler 10% des 'clients' pour atteindre un facteur de lift de 3 vs. le cas sans modèle (la droite rouge = la réponse générale des 100% des clients)
- Et un facteur de 1,5 à 50%.



Génération du Lift Chart (suite)

Exemple 3 de construction de Lift Chart :

• Supposons avoir 10000 instances de test dont 500 sont positives.

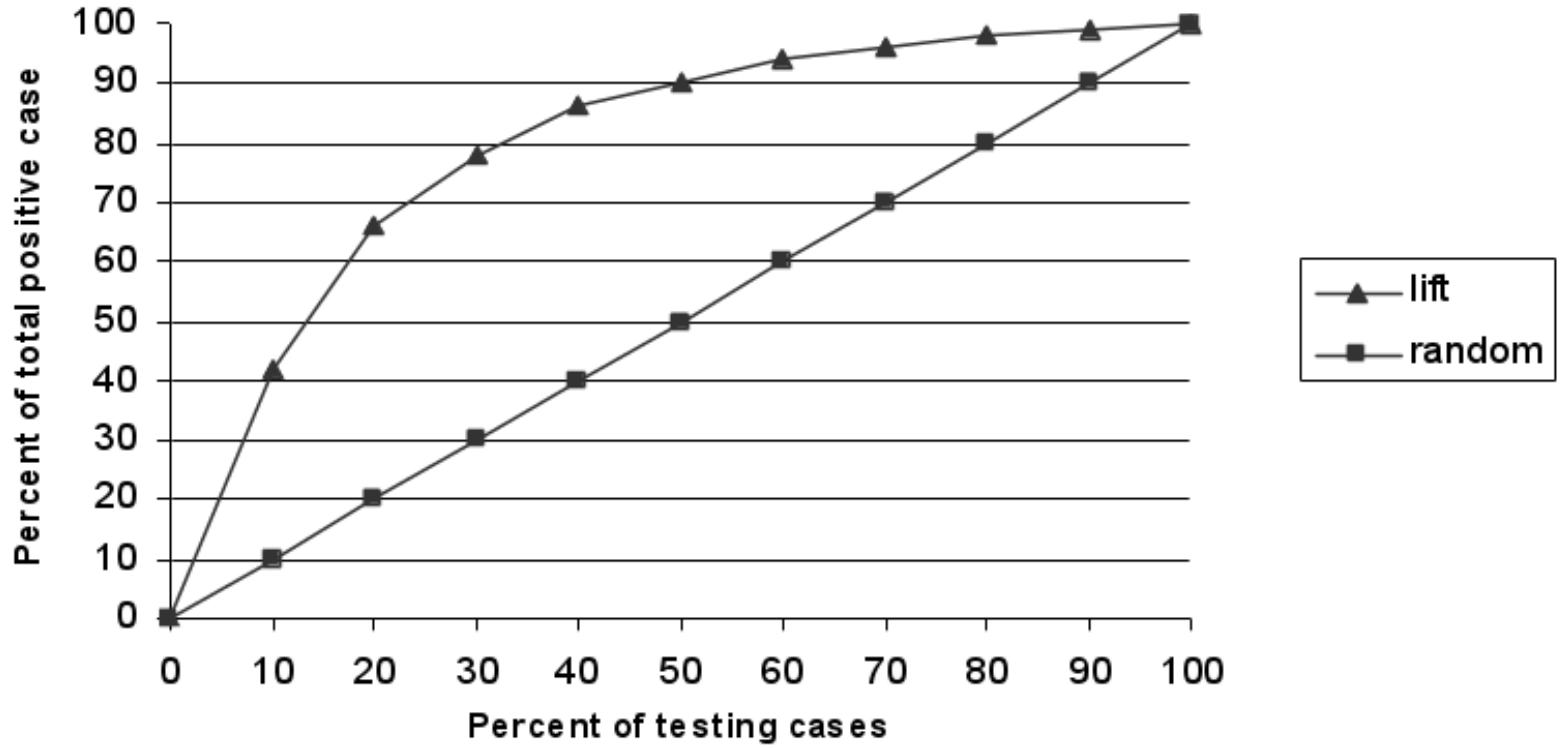
• Ayant obtenu un modèle de prédiction, on établit le score des instances de test puis on crée (par exemple) 10 paquets où chaque paquet a 1000 instances (de test) en ordonnant par les paquets par leur score.

Soit par exemple :

- Le paquet 1 a 210 instances positives
- Le paquet 2 a 120 instances positives
- Le paquet 3 a 60 instances positives
- .....
- Le paquet 10 a 5 instances positives

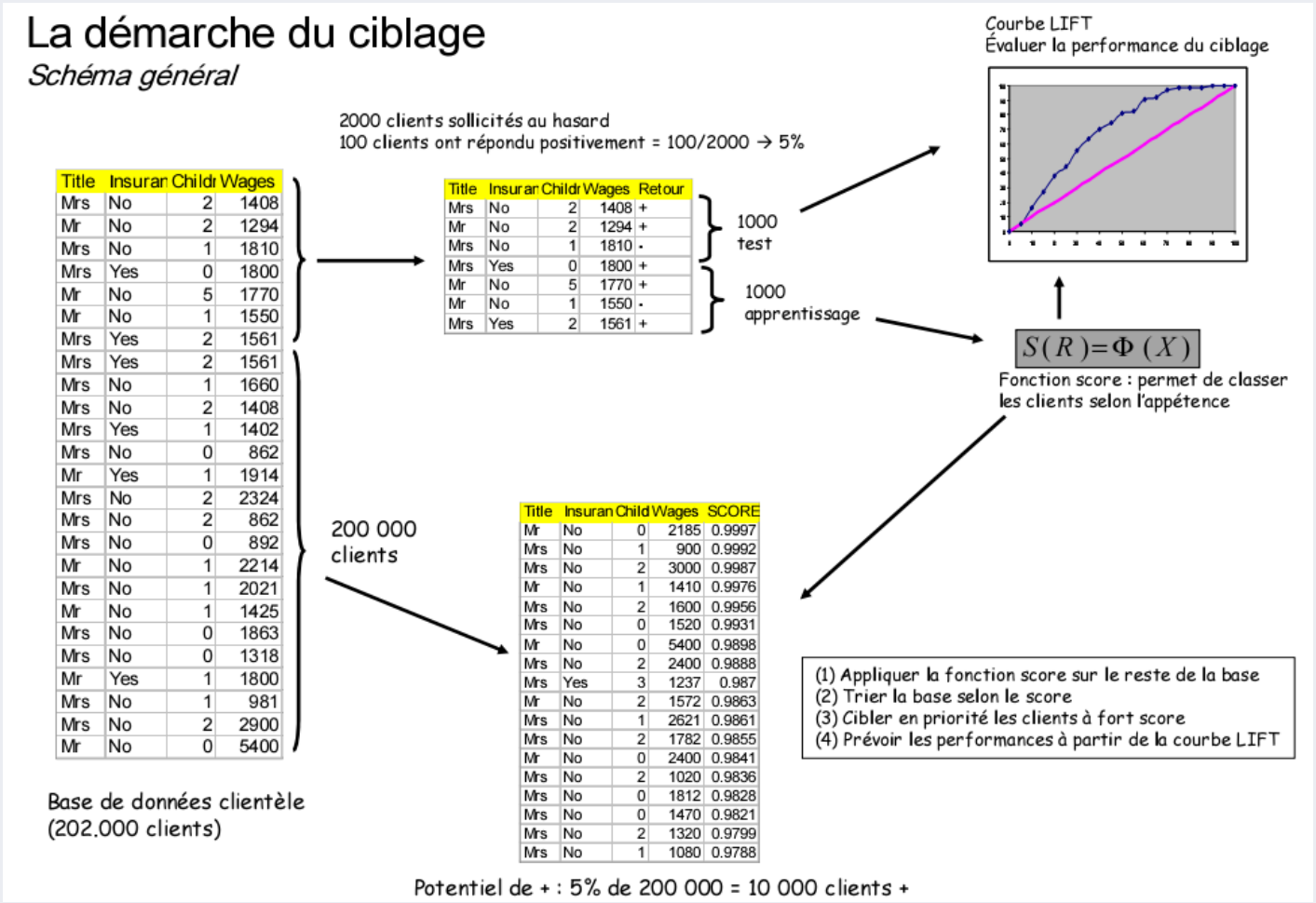
Génération du Lift Chart (suite)

1	2	3	4	5	6	7	8	9	10
210	120	60	40	22	18	12	7	6	5
42%	24%	12%	8%	4.40%	3.60%	2.40%	1.40%	1.20%	1%
42%	66%	78%	86%	90.40%	94%	96.40%	97.80%	99%	100%



Addendum : détails construction de Lift Chart

- Lift avec apprentissage (Thanks to Ricco Rakotomalala) :



Title	Insuran	Childr	Wages
Mrs	No	2	1408
Mr	No	2	1294
Mrs	No	1	1810
Mrs	Yes	0	1800
Mr	No	5	1770
Mr	No	1	1550
Mrs	Yes	2	1561
Mrs	Yes	2	1561
Mrs	No	1	1660
Mrs	No	2	1408
Mrs	Yes	1	1402
Mrs	No	0	862
Mr	Yes	1	1914
Mrs	No	2	2324
Mrs	No	2	862
Mrs	No	0	892
Mr	No	1	2214
Mrs	No	1	2021
Mr	No	1	1425
Mrs	No	0	1863
Mrs	No	0	1318
Mr	Yes	1	1800
Mrs	No	1	981
Mrs	No	2	2900
Mr	No	0	5400

Title	Insuran	Childr	Wages	Retour
Mrs	No	2	1408	+
Mr	No	2	1294	+
Mrs	No	1	1810	-
Mrs	Yes	0	1800	+
Mr	No	5	1770	+
Mr	No	1	1550	-
Mrs	Yes	2	1561	+

Title	Insuran	Child	Wages	SCORE
Mr	No	0	2185	0.9997
Mrs	No	1	900	0.9992
Mrs	No	2	3000	0.9987
Mr	No	1	1410	0.9976
Mrs	No	2	1600	0.9956
Mrs	No	0	1520	0.9931
Mr	No	0	5400	0.9898
Mrs	No	2	2400	0.9888
Mrs	Yes	3	1237	0.987
Mr	No	2	1572	0.9863
Mrs	No	1	2621	0.9861
Mrs	No	2	1782	0.9855
Mr	No	0	2400	0.9841
Mrs	No	2	1020	0.9836
Mrs	No	0	1812	0.9828
Mrs	No	0	1470	0.9821
Mrs	No	2	1320	0.9799
Mrs	No	1	1080	0.9788

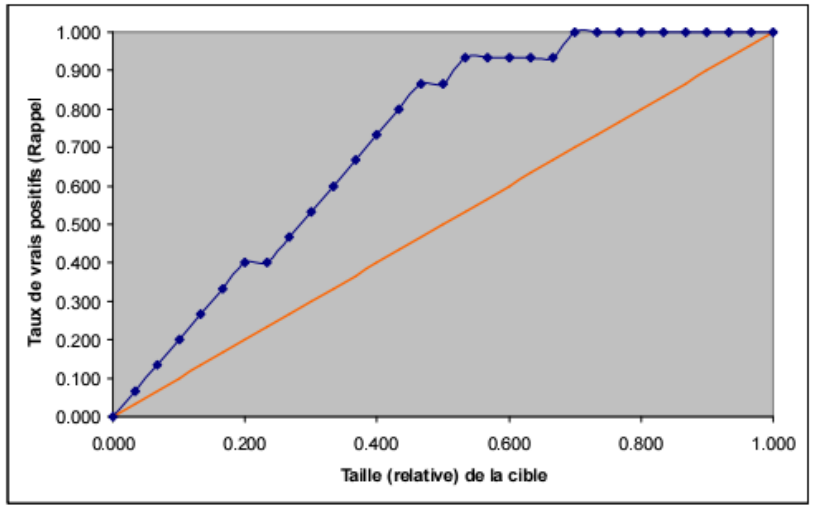
Addendum : détails construction de Lift Chart (suite)

- Lift sur un fichier de données (Thanks to Ricco Rakotomalala) :

Trier les données selon les scores décroissants

i	Retour	Score	Taille Cible	Rappel (TVP)
			0.000	0.000
1	positif	1.000	0.033	0.067
2	positif	1.000	0.067	0.133
3	positif	0.999	0.100	0.200
4	positif	0.999	0.133	0.267
5	positif	0.998	0.167	0.333
6	positif	0.992	0.200	0.400
7	négatif	0.987	0.233	0.400
8	positif	0.987	0.267	0.467
9	positif	0.974	0.300	0.533
10	positif	0.969	0.333	0.600
11	positif	0.953	0.367	0.667
12	positif	0.952	0.400	0.733
13	positif	0.942	0.433	0.800
14	positif	0.825	0.467	0.867
15	négatif	0.772	0.500	0.867
16	positif	0.590	0.533	0.933
17	négatif	0.507	0.567	0.933
18	négatif	0.307	0.600	0.933
19	négatif	0.294	0.633	0.933
20	négatif	0.109	0.667	0.933
21	positif	0.073	0.700	1.000
22	négatif	0.035	0.733	1.000
23	négatif	0.024	0.767	1.000
24	négatif	0.016	0.800	1.000
25	négatif	0.015	0.833	1.000
26	négatif	0.009	0.867	1.000
27	négatif	0.004	0.900	1.000
28	négatif	0.003	0.933	1.000
29	négatif	0.002	0.967	1.000
30	négatif	0.000	1.000	1.000

N	30
N(positif)	15

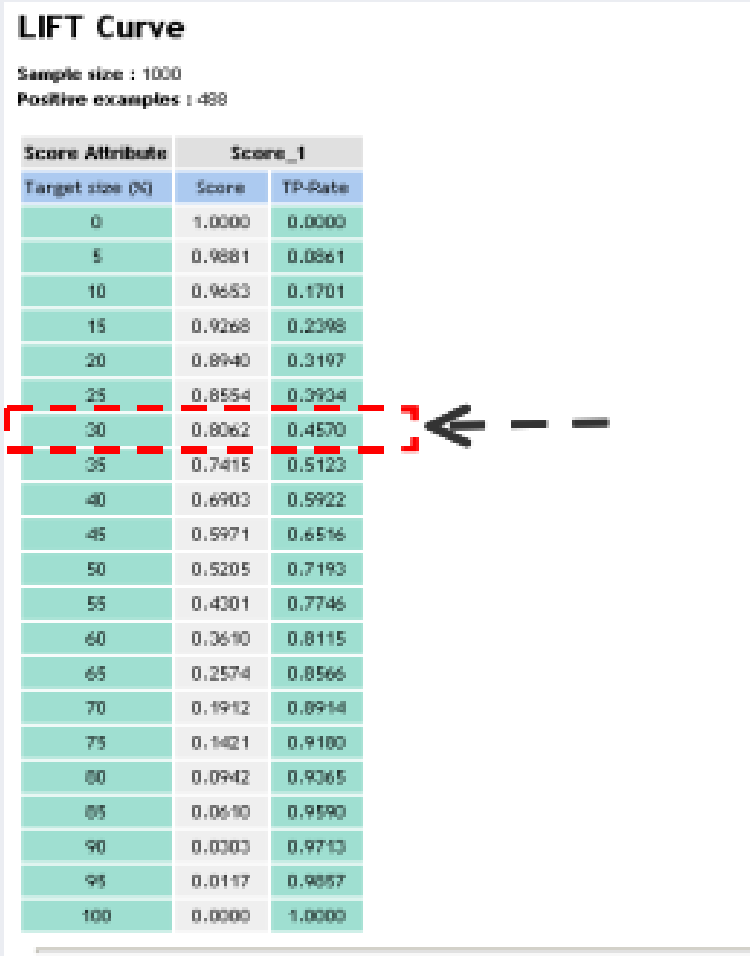


Taille relative de la cible cumulée = i / N

TVP = N(positifs parmi les i premiers) / N(positifs)

Addendum : détails construction de Lift Chart (suite)

Exemple 2 : Régression dans une étude publicitaire :



LIFT Curve

Sample size : 1000
Positive examples : 488

Score Attribute	Score_1	
Target size (%)	Score	TP-Rate
0	1.0000	0.0000
5	0.9881	0.0861
10	0.9653	0.1701
15	0.9268	0.2398
20	0.8940	0.3197
25	0.8554	0.3934
30	0.8062	0.4570
35	0.7415	0.5123
40	0.6903	0.5922
45	0.5971	0.6516
50	0.5205	0.7193
55	0.4301	0.7746
60	0.3610	0.8115
65	0.2574	0.8566
70	0.1912	0.8914
75	0.1421	0.9180
80	0.0942	0.9365
85	0.0610	0.9590
90	0.0303	0.9713
95	0.0117	0.9857
100	0.0000	1.0000

- Sur les 1000 individus en test, 488 sont positifs.
- Si on cible 300 individus (30% de 1000), on pourra espérer atteindre 46% des positifs,
  soit $46\% \times 488 = 225$ individus.
- Si on avait envoyé les lettres au hasard, sans ciblage, on aurait obtenu $30\% \times 488 = 146$ réponses positives.

➛ Le gain : (225 - 146) = 79 individus supplémentaires conquis.

Diagramme de Gain

- Très proche de Lift Chart
- Calcule le ratio entre l'apport du modèle et le cas sans modèle (aléatoire)
- Par opposition à la matrice de confusion qui considère l'ensemble de la *population*, les diagrammes de Lift ou de Gain s'intéresse au gain sur une partie de cette population.

Exemple de tables construites :

Score Table

Target	Score
0	235
1	724
1	556
0	345
0	480
1	676
0	195
1	880
0	368
...	...

Sorted by Score

Target	Score
1	880
1	724
1	676
1	556
0	480
0	368
0	345
0	235
0	195
...	...

Gain Table

Count%	Target%
10	36
20	54
30	66
40	76
50	85
60	90
70	94
80	98
90	100
100	100

Lift Table

Count%	Lift
10	3.6
20	2.7
30	2.2
40	1.9
50	1.7
60	1.5
70	1.3
80	1.2
90	1.1
100	1

Diagramme de Gain (suite)

Score Table

Target	Score
0	235
1	724
1	556
0	345
0	480
1	676
0	195
1	880
0	368
...	...

Sorted by Score

Target	Score
1	880
1	724
1	676
1	556
0	480
0	368
0	345
0	235
0	195
...	...

Gain Table

Count%	Target%
10	36
20	54
30	66
40	76
50	85
60	90
70	94
80	98
90	100
100	100

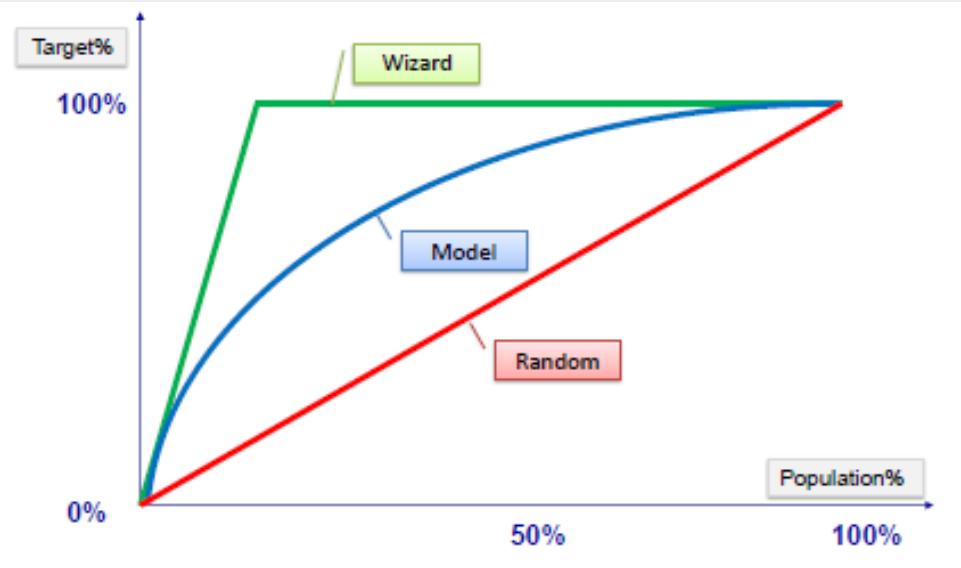
Lift Table

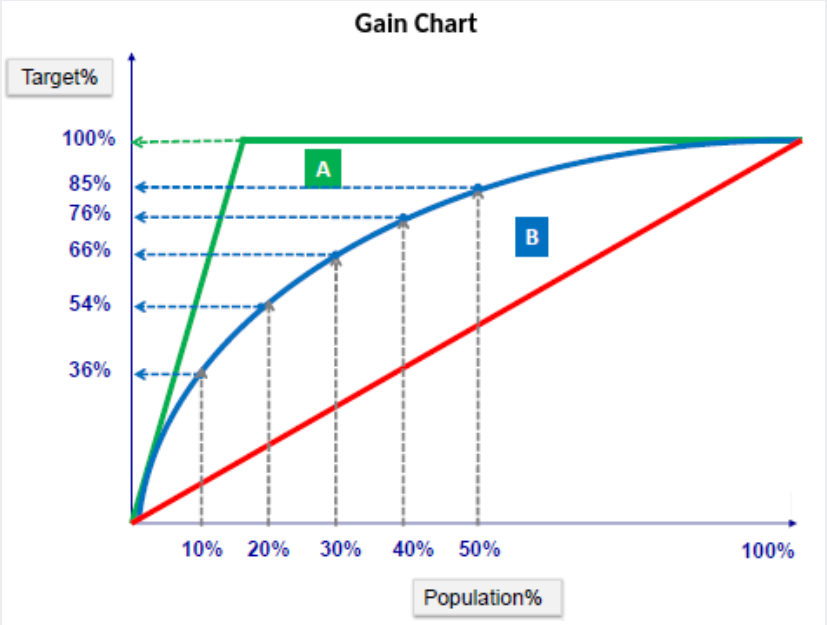
Count%	Lift
10	3.6
20	2.7
30	2.2
40	1.9
50	1.7
60	1.5
70	1.3
80	1.2
90	1.1
100	1

- Table gauche : un score (probabilité de TP) est associé à chaque instance (TP uniquement)
- 2e table : on trie selon le score
- Table 3 : la table des gains
- Table 4 : table Lift équivalente.

Diagramme de Gain (suite)

Gain chart :





- A gauche, ce que l'on voudrait (l'idéal = 100%)
  Le modèle donne la courbe B (données d'apprentissage), le rouge le hasard.
- A droite : Diagramme de Gain annoté.
  A l'aide du modèle, on atteint un taux de 85% en visant 50% de la population le plus prometteur.

La courbe ROC et AUC

Un exemple : comparaison (points de ROC) de 4 matrices de confusion

A

TP=63	FP=28	91
FN=37	TN=72	109
100	100	200

TPR = 0.63
FPR = 0.28
PPV = 0.69
F1 = 0.66
ACC = 0.68

B

TP=77	FP=77	154
FN=23	TN=23	46
100	100	200

TPR = 0.77
FPR = 0.77
PPV = 0.50
F1 = 0.61
ACC = 0.50

C

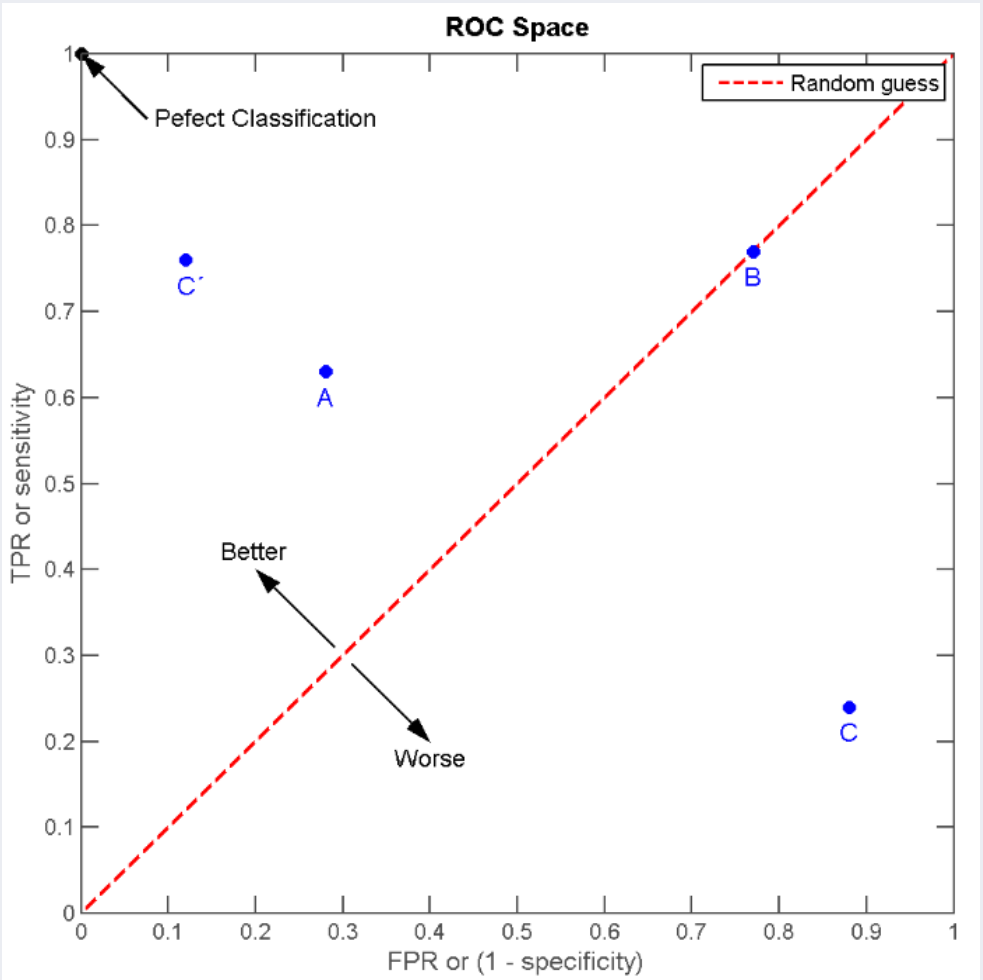
TP=24	FP=88	112
FN=76	TN=12	88
100	100	200

TPR = 0.24
FPR = 0.88
PPV = 0.21
F1 = 0.22
ACC = 0.18

C'

TP=76	FP=12	88
FN=24	TN=88	112
100	100	200

TPR = 0.76
FPR = 0.12
PPV = 0.86
F1 = 0.81
ACC = 0.82



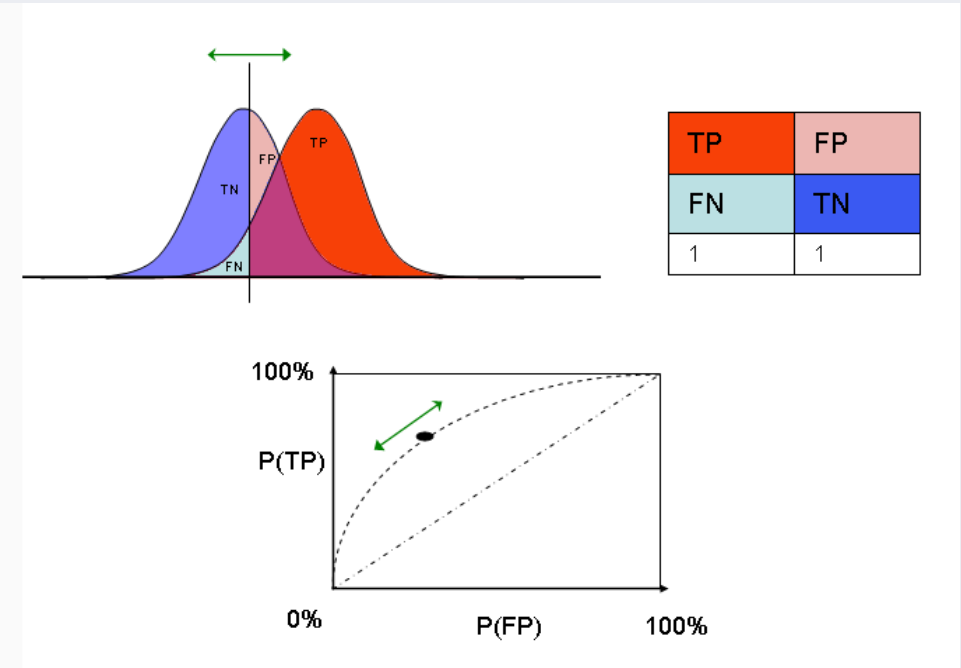
- PPV= précision= $\frac{TP}{TP+FP}$
- F1= $\frac{2TP}{2TP+FP+FN}$
- ACC= accuracy= $\frac{TP+TN}{P+N}$
- FP=(1-spécificité)=1-TN (axe Xs)

La courbe ROC et AUC (suite)

La Courbe de ROC (*Receiver operating characteristic*) :

- En classification, on manipule (souvent) des variables aléatoires $\in \mathbb{R}$.
- Soit $P_1(T)$ la probabilité d'appartenir à une classe C :
  $\rightarrow$ une fonction paramétrée par une décision (ou un seuil de discrimination) T et
- Soit $P_0(T)$ la probabilité de NE PAS appartenir à C, ☞ v. $\sum = 1$ dans cols mat. conf.

- Le taux False Positive :
$$FPR(T) = \int_T^{\infty} P_0(T)dT$$
- Le taux True Positive :
$$TPR(T) = \int_T^{\infty} P_1(T)dT.$$
- En faisant varier T, la courbe ROC représente $TPR(T)$ versus $FPR(T)$.
- Le coin haut gauche représente un classifieur PARFAIT.
- Le polynôme en pointillé mesure le comportement du classifieur.
- La droite en pointillé $y = x$ représente le **Hasard**.



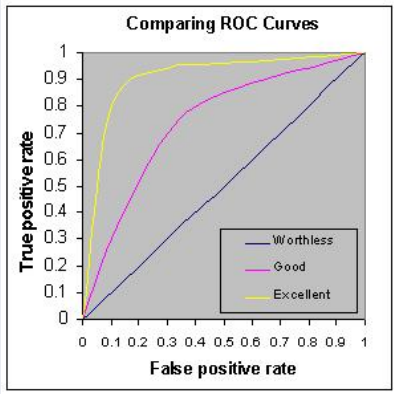
- Dans la fig. ci-dessus : le seuil (la barre verticale entre les 2 cloches) et sa *pente* repérée par $\leftrightarrow$ est reportées sur la courbe ROC (gros point noir).

La courbe ROC et AUC (suite)

- En générale, on considère l'aire sous la courbe ROC (AUROC) :

  [0.90, 1] = Excellent (A),
  [0.8, 0.9] = Bon (B),
  [0.7, 0.8] = Assez bon (C),
  [0.6, 0.7] = Faible (D),
  [0.5, 0.6] = Échec (F).

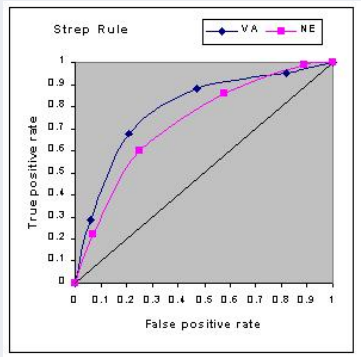
  ➙ Rappelons que sous 0.5, on fait pire qu'une Pile-ou-Face.



- **Exemple** (thanks to http ://gim.unmc.edu) :

Ici, deux courbes ROCs sur des études cliniques de deux régions aux USA : *Viginia* (VA) et *Nebraska* (NE) sont comparées.

Parmi ces deux modèles, celui de VA (avec AUROC=0.78) est considéré meilleur que celui de NE (AUROC=0.73)



- L'AUROC mesure la capacité de *discrimination* d'un modèle sur les données de l'ensemble de test : séparer les TP des TN (e.g. malade ou pas).
- **Si on sélectionne au hasard deux instances de test, plus l'AUC est élevée, mieux le modèle devrait en désigner les bonnes classes (les discriminer sans se tromper).**
- La courbe ROC permet principalement de comparer deux modèles, indépendamment de leurs coûts de mauvaise affectation.
- Construction de ROC : même principe que pour Lift

La courbe ROC et AUC (suite)

- L'AUROC est le pourcentage des paires d'instances aléatoires de test bien classées par le modèle.
- Une courbe ROC exprime les points suivants :
  - le rapport entre la sensitivité et la spécificité du modèle :
    ➙ si la sensitivité augmente, la spécificité diminue.
  - plus la courbe colle à l'axe vertical avant de se coller à l'axe horizontale (supérieur), mieux sera le modèle.
  - *a contrario*, plus cette courbe est proche de la diagonale, moins les résultats sont meilleurs que le hasard.
  - la pente de la tangente à un point donnée de la courbe donne le taux de *vraisemblance* (LR) de ce point.
    ➙ Pour calculer (trivialement) cette pente, on choisit deux points (x_1, y_1) et (x_2, y_2) (par exemple sur la courbe et sur l'axe croisé par la tangente) : la $pente = \frac{y_2 - y_1}{x_2 - x_1}$

☞ La courbe ROC vient de la théorie de détection du signal ; elle a été créée pendant la 2e guerre mondiale où un opérateur de radar avait besoin de savoir (discriminer) si un point détecté sur le radar désignait un ami, un ennemie ou un bruit. Cette capacité a pris le nom "Receiver Operating Characteristics" qui perdure avec un large domaine d'utilisation (en prédiction, en particulier dans le domaine médical depuis 1970).

Définition / estimation de l'AUC

• En utilisant des valeurs normalisées, l'aire sous la courbe est la **probabilité** qu'un classifieur classe une instance aléatoire POSITIVE mieux qu'une NÉGATIVE

(en supposant qu'une instance positive est classée plus haut qu'une négative)

Dans ce cas, on obtient la valeur de l'AUC ($FPR'(T)d(T)$: dérivée de FPR).

$$AUC = \int_{\infty}^{-\infty} TPR(T)\ \partial FPR(T)\ \partial T \quad \text{ou} \quad \int_{\infty}^{-\infty} TPR(T).FPR'(T)dT$$

$$= \int_{\infty}^{-\infty} TPR(T)P_0(T)$$

Rappel : $TPR = \frac{TP}{TP + FN}$

➙ AUC est trivialement exprimée par $\int_{\infty}^{-\infty} f(t)\partial t$ de la fig. courbe de ROC.

➙ TPR représente la moyenne de la distribution des instances positives.

➙ P_0 est la probabilité de ne pas appartenir à la classe (concernée par TPR).

➙ Si $FPR(T) = \int_{\infty}^{-\infty} P_0(T)dT$ alors sa dérivée est bien $P_0(T).d(T)$

☞ Bornes inversées car un grand T a une plus petite valeur sur l'axe des Xs.
P. Ex. : on dira qu'un score $s < T$ donne un classement positif (par le modèle)
et $s \geq T$ donne un classement négatif.

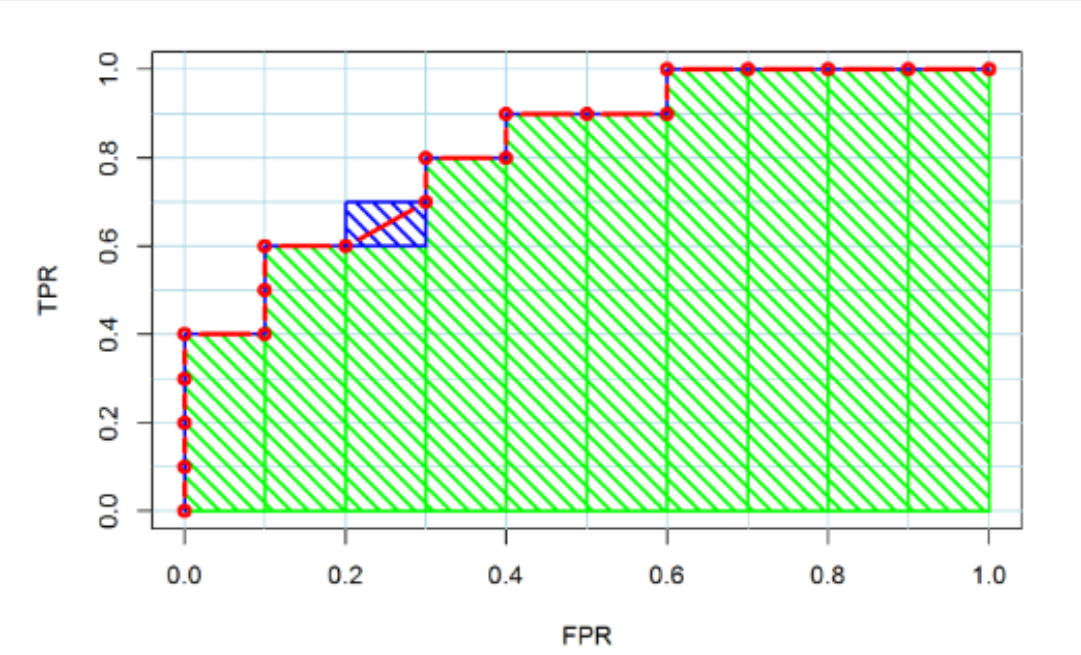
• L'idéal : l'aire maximum sous la courbe (le plus éloigné du Hasard).

Définition / estimation de l'AUC (suite)

Comment mesurer l'AUC ?

- On peut calculer (géométrie) l'aire sous la courbe en approximant des rectangles ($TP \times FP$) obtenus / utilisés lors de la création du ROC.

• Le rectangle bleu peut être ajouté à AUC pour palier les erreurs éventuelles de calculs.



- On peut <u>estimer</u> cette aire.

☞ MCL possible.

Définition / estimation de l'AUC (suite)

Estimation (par exemple) avec MCL :

Pour estimer AUC, on s'appuie sur la définition même de l'AUC :

- AUC est la proba (%) de paires d'instances biens classées par le modèle
- Pour un cas bi-classes, soit n_1 observations positives (x^+) (classées dans TP) et n_2 observations négatives (pas dans TP, noté x^-).
- En général, dans un modèle utile, $score(x^+) > score(x^-)$ et
$$AUC = Pr(score(x^+) > score(x^-))$$
- On a $t = n_1.n_2$ nombre de paires possibles et $n = n_1 + n_2$
- Pour une paire d'observations (x^+, x^-), $x^+ \in TP, x^- \notin TP$, si la probabilité (donnée par le modèle) de x^+ est supérieur à celle de x^-, on dira que (x^+, x^-) est une **paire concordante**.
- Soit nc paires concordantes, nd paires discordantes et $(t - nc - nd)$ paires ex-aequo.

➛ $AUC = \frac{nc}{t}$ sera le pourcentage des paires concordantes (parmi les t paires possibles)

Définition / estimation de l'AUC (suite)

Exploitation du ROC

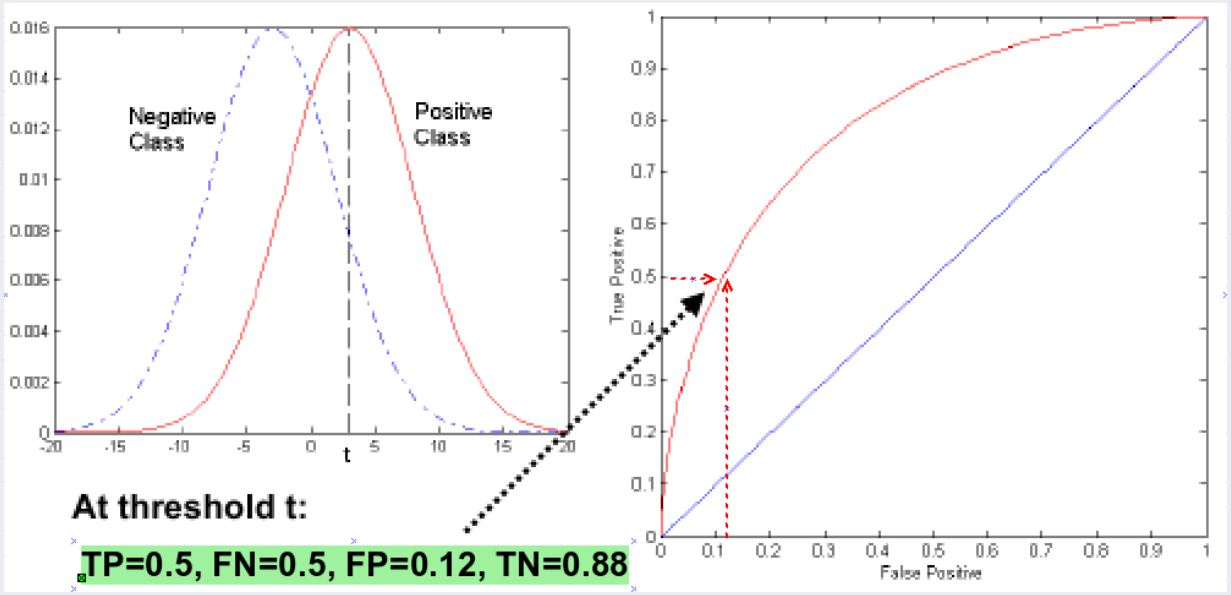


FIGURE 3: Courbe ROC pour un cas à 2 classes

- **Tout point $x > t$ (t = le seuil) est considéré comme Positif.**

ROC et Arbres de décision

Courbe ROC pour les arbres de décision (cf. Weka) :

• La méthode J48 génère des probabilité de classe basée sur la fréquence relative observée des classes au niveau de chaque feuille.

➛ On établit ainsi une liste pour les instances avec ces valeurs

• Pendant la construction de l'arbre de décision, Weka collecte les fréquences au niveau des feuilles (et des noeuds internes). Ces fréquences correspondent à la valeur de la classe de ces instances d'apprentissage qui descendent dans ces feuilles.

➛ Ces valeurs sont ensuite normalisées pour devenir des probabilités.

• Sachant les "vraies" classes, on peut, pour chaque instance de cette liste, calculer la valeur TP (ainsi que les autres valeurs statistiques).

• On peut ensuite utiliser ces estimations pour l'ensemble de test pour la classe positive (e.g. yes), imposer un seuil et obtenir un point de la courbe ROC.

• Exemple au niveau d'une feuille ../..

ROC et Arbres de décision (suite)

• Par exemple, si on a 10 instances réparties dans deux feuilles (7 et 3) et les nombres des "positives" sont 4 et 2. Chacune de ces valeurs sera divisée ensuite par le nbr d'instances arrivées sur ces feuilles.

• Par exemple, on peut avoir (pour la branche à 7 instances dont 4 positives) :

1. 20% -
2. 33% -
3. 45% +
4. 55% +
5. 67% +
6. 69% -
7. 80% +

Question de seuil :

• En général, un seuil de 50% est considéré (la diagonale)

• Les couts (de FP et de FN) sont en général pas connus.

• Mais si on peut avoir une information sur le coût d'une mauvaise classification, on peut tout simplement l'ajouter (pour les FP et FN) dans la ROC.

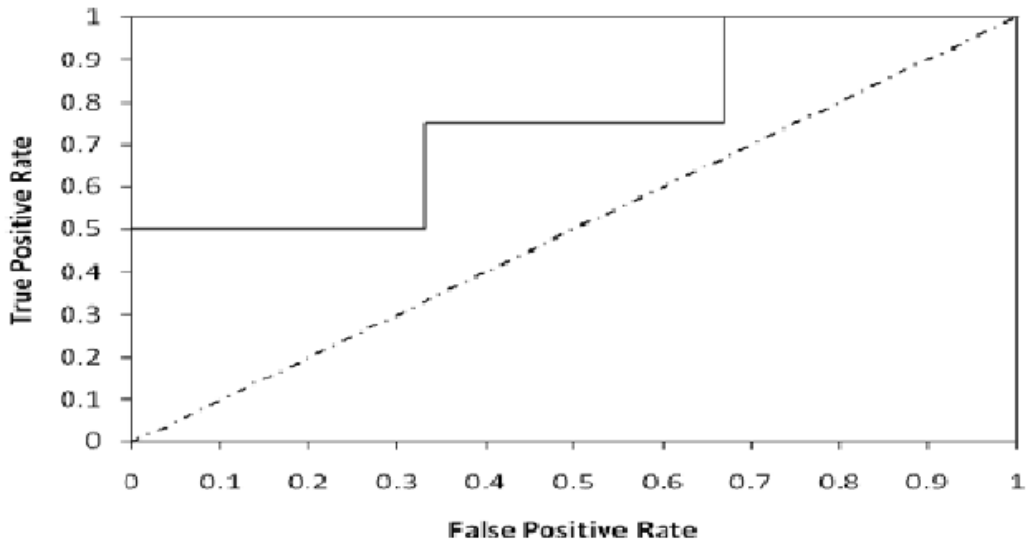
➛ Le point avec le coût minimal sera le seuil de minimisation de coût.

☞ Rappel : $TPR = \frac{TP}{TP + FN}$ obtenu par la matrice de confusion.

ROC et Arbres de décision (suite)

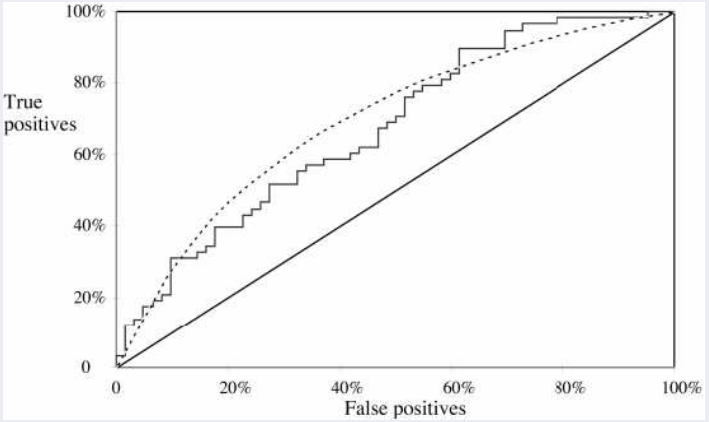
Un exemple récapitulatif de construction de la courbe ROC :

Rank		1	2	3	4	5	6	7	8	9	10
Actual class		+	+	–	–	+	–	–	+	–	–
TP	0	1	2	2	2	3	3	3	4	4	4
FP	0	0	0	1	2	2	3	4	4	5	6
TN	6	6	6	5	4	4	3	2	2	1	0
FN	4	3	2	2	2	1	1	1	0	0	0
TPR	0	0.25	0.5	0.5	0.5	0.75	0.75	0.75	1	1	1
FPR	0	0	0	0.17	0.33	0.33	0.50	0.67	0.67	0.83	1



ROC VS X-V

- La ligne zigzaguée dépend des détails de l'échantillon de l'ensemble de test .
- Pour réduire cette dépendance, on utilise une X-validation.
- Pour chaque nombre différent de FP ("Non" sur l'axe horizontal) :
  - Prendre assez d'exemples dans la liste ordonnée qui incluent ce nombre de "Non" et compter le nombre de "Oui" que cela contient.
  - Faire la moyenne de ce nombre sur différents plis de la X-validation.
  - Le résultat sera la courbe lisse de la figure



- La courbe ROC montre les performances d'un classifieur sans regarder la distribution des classes, ni le coût des erreurs .

- Méthode implantée dans **Weka**
  ➙ Weka propose d'autres courbes de coût et d'erreur

ROC VS X-V (suite)

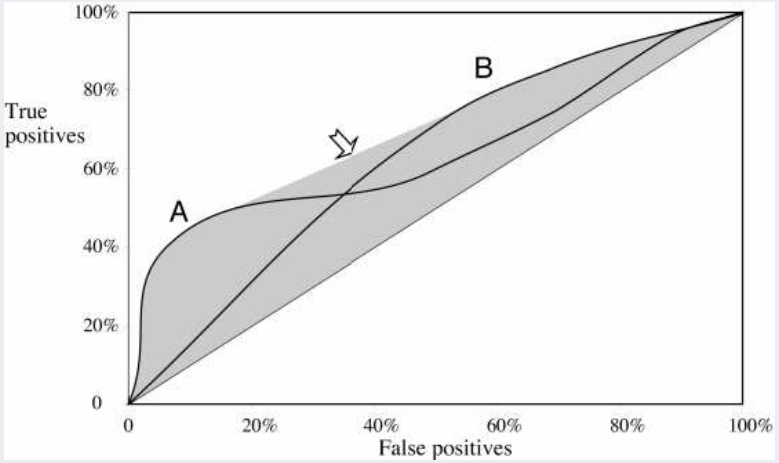
Détails Courbe ROC et X-validation :

• Pour chaque position sur l'axe horizontal, trouver le point correspondant sur l'axe vertical en utilisant Ten-fold-X-validation où la quantité à calculer (avec sa moyenne) sur les 10 plis est le nombre de "Oui" (réponse positive) dans un échantillon contenant le nombre approprié de "Non".

• Pas besoin de répéter le Ten-fold-X-validation encore et encore pour toute position horizontale : étant donné la sortie d'un One-fold de la X-validation (une liste ordonnée des probabilités des instances), on peut lire la position verticale pour toute position horizontale juste en enregistrant le nombre de "Oui" dans les échantillons contenant chaque nombre possible de "Non".

• Faire la moyenne de ces schémas sur les 10 plis $\simeq$ dessiner le nombre de "Oui" contre le nbr. de "Non" ; on dessinera des bandes verticales correspondant à différents nbr. de "Non" et en conservant la moyenne des points dans chaque bande verticale.

ROC de 2 modèles



- La méthode A est excellente sur un petit sous-ensemble bien ciblé, i.e. on travaille vers la partie gauche de la courbe.
  - ➙ Si l'on veut couvrir juste 40% de TP, on choisit la méthode A.
  - ➙ *A* donne un taux de FP d'environ 5% (pour les 40% de TP sur l'axe Y),
  - ➙ B (moins bien) donne env. 30% de FP (pour les mêmes 40% de TP).
- Mais la méthode B est excellente si l'on travaille sur un large échantillon :
  - Si l'on veut avoir 80% de TP, B donnera un taux de 60% de FP (vs. 80% pour A).
- La zone grisée est appelée *fuselage convexe* (hull) des 2 courbes :
  - ➙ On prend toujours un point sur la **frontière supérieure** de ce fuselage.

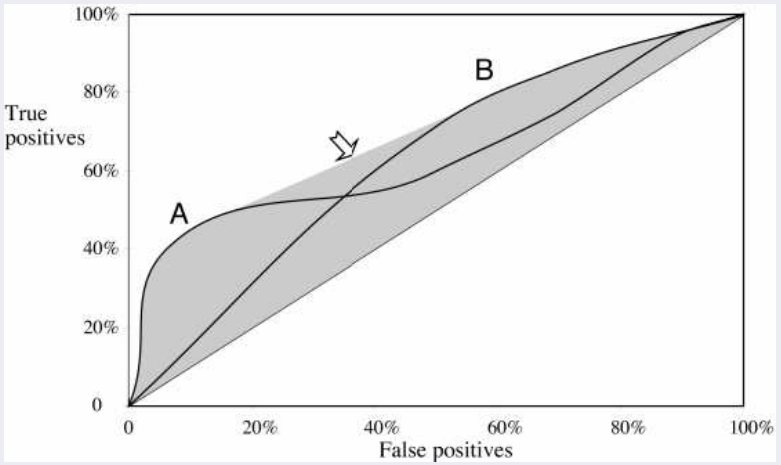
ROC de 2 modèles (suite)

• **La région grisée au milieu (ni A ni B) du fuselage :**

◦ On peut se promener dans cette région en combinant les 2 méthodes

◦ On choisit une partie de la méthode A qui donne les taux TP et FP resp. t_A et f_A,
➞ Idem pour B donnant resp. t_B et f_B.

◦ Si on utilise ces 2 schémas de manière aléatoire avec des probabilités p et q $(p + q = 1)$:
➞ On aura les taux TP et FP : $p.t_A + q.t_B$ et $p.f_A + q.f_B$.



• Ceci représente un point sur la droite joignant les points $(t_A\ ,\ f_A)$ et $(t_B\ ,\ f_B)$
➞ En variant p et q, on peut tracer toute la ligne entre ces deux points.

• Par ce moyen, toute la région grisée peut être atteinte.

Si un schéma particulier génère un point (même un seul) sur le fuselage convexe,
➞ On prend ce schéma

Sinon, utiliser une combinaison de classifications donnant un point du fuselage.

Addendum : ROC

- **ROC** : "Receiver Operating Characteristic"
- Habituellement utilisé en détection/traitement de signal :
  ➙ Établit le rapport entre le taux de succès (hit rate) vs. fausses alertes sur un canal ébruité.
- **Très proche de Lift Chart** (les mêmes conditions de choix d'échantillon)

<u>Différences par rapport à Lift Chart :</u>
∘ L'axe Y : pourcentage de **TP** dans un échantillon (au lieu du nbr. TP)
  ➙ Si le % très petit (e.g. 0,1% en Marketing)
  ➙ on prend le total TP à la place.
∘ L'axe X : pourcentage de **FP** dans un échantillon (au lieu de % de la taille de l'échantillon)

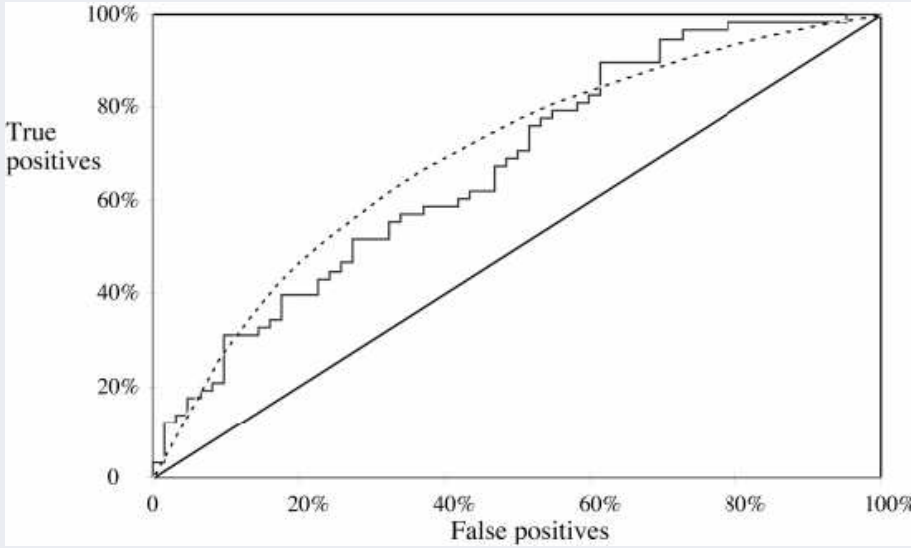
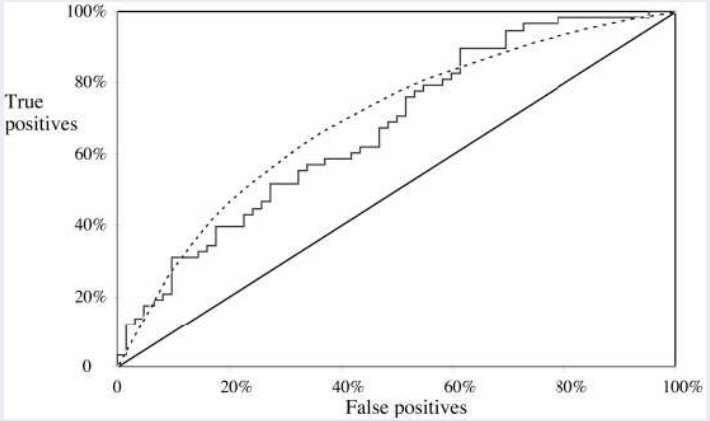


TABLE 6: Un exemple de courbe ROC

Addendum : ROC (suite)

Exemple de courbe ROC :

- **Coin (0,0)** : tout est considéré comme négatif,
- **Coin (1,1)** : tout positif
- **Coin (1,0)** : idéal
- Sous la diagonale : mauvaise prédiction
  = l'inverse de la vraie classe (voir axes)



Pour dessiner le zigzag (pour la table de l'ex. *Lift Chart* rappelée ci-dessous) :

- axe X = %FP, axe Y = %TP
- De l'origine, aller 2 fois vers le haut (2 TP), puis un horizontal (1 FP), 5 positifs (vers le haut), puis ...
- Chaque point correspond au dessin d'une ligne à une certaine position dans la liste numérotée des instances.

Rappel 100 instances, taux de succès = 20% (le non de la 3e ligne est un FP)

No	Probabilité Prédite	Vraie classe	No	Probabilité Prédite	Vraie classe
1	0.95	oui	...	...	...
2	0.93	oui	9	0.80	non
3	0.93	non	10	0.79	oui
4	0.88	oui	11	0.77	non
...	...	...	...	...	...

Complément : Intégrer le coût à un schéma DM

- La plupart des schémas d'apprentissage ne tiennent pas compte des coûts
  ➙ Génèrent le même classifieur indépendamment du coût de chaque classe.
- Certains schémas peuvent en tenir compte, en utilisant les probas :
  ➙ Bayes Naïve peut en tenir compte via les **probabilités**
  ➙ Dans un arbre de décision , la distribution de probabilités est celle des instances au niveau des feuilles.
- N.B. : un arbre de décision peut être utilisé pour dessiner une courbe ROC
  ➙ mais l'information sur le coût n'est pas utilisée pendant l'apprentissage .
- Ajuster un schéma d'apprentissage aux coûts = améliorer performances

Complément : Intégrer le coût à un schéma DM (suite)

Comment peut-on faire pour intégrer le coût ?

• Un moyen simple et général qui rend TOUT schéma d'apprentissage sensible au coût (situation **à 2 classes**).

• **Idées :**
  ➙ Générer des échantillons avec différentes proportions de "Oui"/"Non".
  ➙ Ré échantillonner (ou pondérer) les instances selon le coût
  ➙ Augmenter artificiellement (dupliquer) les instance d'une classe donnée
  ➙ etc...

• Pour construire des arbres de décision sensibles au coût :
  ➙ initialiser l'arbre avec les coûts relatifs de 2 types d'erreur FP et FN

• Beaucoup d'outils DM permettent de pondérer les instances.

Complément : Intégrer le coût à un schéma DM (suite)

Exemple : on augmente artificiellement le nombre de "Non" (FN) par un facteur (par exemple de 10) et on applique ensuite l'apprentissage .

- Le Schéma cherche à diminuer l'erreur
  - ➙ produira une structure qui évite les erreurs sur les instances avec "Non" (FN)
  - ➙ une erreur de ce type est pénalisée 10 fois plus.
- Tester sur la DB originale, on aura :
  - ➙ MOINS d'erreur sur les FN ("Non", 10 fois plus) que sur les "Oui" (FP)

• **Liens avec les valeurs manquantes** : construire des arbres de décision en **éclatant** une instance en pièces et envoyer les *morceaux de l'instance* dans chaque branche.

➙ On éclate en utilisant une pondération numérique (poids par défaut=1).

Complément : Intégrer le coût à un schéma DM (suite)

• **Une technique générale** pour un apprentissage de **Courbe ROC** sensible au coût :

Pour chaque pli de la Ten-fold-X-validation :
- ◦ pondérer les instances par une sélection de différents ratio de coût,
- ◦ entraîner le schéma sur chaque sous-ensemble pondéré,
- ◦ compter les TP et FP dans l'ensemble de test (les 2 axes de ROC),
- ◦ calculer le point correspondant sur la courbe ROC, peu importe si l'ensemble de test est pondéré ou non (les axes de ROC sont exprimés en pourcentage de TP et FP)
- ◦ dessiner la moyenne des points dans chaque bande verticale et créer ROC.

• Cette technique n'impose pas de contrainte sur le schéma d'apprentissage .

• Problème : plus coûteux que pour les classifieurs qui intègrent le coût
➙ il faut une Ten-fold-X-validation séparée pour chaque point de ROC.

Autres courbes de mesures

• Le Lift-chart et ROC également utilisés dans divers domaines, e.g. en Recherche d'Information (IR)

• Pour une requête donnée, un outil de recherche (e.g. sur WEB) produit une liste de succès (hits) = les documents supposés **relevants** de la requête.

• Lequel est mieux ? un système qui localise 100 documents dont 40 sont relevants OU un autre qui localise 400 dont 80 sont relevants.

• La réponse dépend du coût relatif de **FP** (*les documents extraits qui ne sont pas relevants*) et **FN** (*documents qui sont relevants mais pas extraits*).

• Comme en DM, on utilise en IR 2 mesures (**en phase de test**) : **Recall** et **Précision**.

$$\textbf{recall} = \frac{\textit{nombre de docs extraits qui sont relevants (TP)}}{\textit{total des docs relevants (TP + FN potentiels)}}$$

$$\textbf{precision} = \frac{\textit{nombre de docs extraits qui sont relevants (TP)}}{\textit{total des docs extraits (TP+TN)}}$$

Autres courbes de mesures (suite)

Rappel :

$$\textbf{recall} = \frac{\#\ relevants\ extraits\ (TP)}{total\ relevants\ (TP + FN)}$$

$$\textbf{precision} = \frac{\#\ relevants\ extraits\ (TP)}{total\ extraits\ (TP + FP)}$$

Auxquelles on ajoute en générale **F-mesure** : $= \frac{2 \times recall \times precision}{recall + precision}$

- Remarque sur TP+FP du dénominateur de *précision* :
  ➙ on extrait ce que l'on croît relevant (inclue des TP et FP ; les TN et FN pas extraits)

- **Exemple** : soit la liste ordonnée des "Oui" et des "Non" de la table Lift Chart vue ci-dessus :
  - Une liste des documents extraits + indication s'ils sont ou pas relevants
  - Hyp. : le corpus contient un total de 40 documents relevants (pas tous extraits) :
    ➙ **recall at 10 :** $\frac{8}{40} = 20\%$ $\Rightarrow$ recall des **10** premiers de la liste (8 TP sur 40)
    ➙ **précision at 10 :** $\frac{8}{10} = 80\%$ $\Rightarrow$ 8 relevants sur les **10** considérés.

- N.B. : *recall=complétude* et *précision = justesse.*

Autres courbes de mesures (suite)

• En extraction d'information , on utilise des courbes *recall-precision* en dessinant l'une contre l'autre, pour différents nombre d'informations (de documents) extraites. On s'intéresse également (cf. ROC) à l'aire sous ses courbes.

• Comme pour ROC et Lift Chart : les axes sont différents, les courbes sont hyperboliques et le point intéressant du coté du coin haut droit.

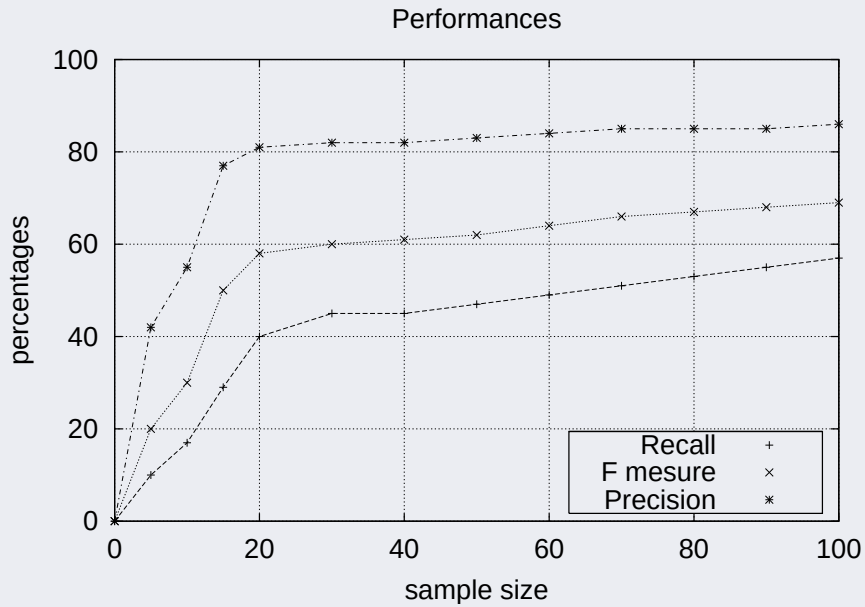


FIGURE 4: Un exemple de courbes Recall/Precision/F-mesure $\left(= \frac{2 \times recall \times precision}{recall + precision}\right)$ en IR

Autres courbes de mesures (suite)

- **Résumé** des 3 différentes façons d'évaluation :

	Domaine	**Type Dessin**	**Axes**	**Explication des axes**
Lift Chart	Marketing	TP vs. la Taille du ss-ensemble	TP Taille ss-ens	Nombre de Vrai Positifs $\frac{TP+FP}{TP+FP+TN+FN} \times 100\%$
Courbe ROC	Communication	**taux TP** vs. taux FP	taux TP taux FP	$\frac{TP}{TP+FN} \times 100\%$ $\frac{FP}{FP+TN} \times 100\%$
Courbes Recall -Precision	IR = Recherche d'information (e.g. Google)	Recall vs. Precision	Recall (voir taux TP) Precision	$\frac{TP}{TP+FN} \times 100\%$ $\frac{TP}{TP+FP} \times 100\%$

TABLE 7: Mesures différentes pour évaluer les résultats faux-positifs vs. Faux-négatifs

Autres courbes de mesures (suite)

Parfois, on cherche une (seule) mesure pour caractériser les performances :

- **En IR :** on utilise souvent
  - ➙ la *moyenne de recall de 3 points* qui donnera la précision moyenne obtenue pour les valeurs de recall at 20, 50 et 80%,
  - ➙ Et la *moyenne de recall de 11 points* (11-pt-average recall) qui donne la précision moyenne pour les recall at 10, 20, 30, 40, 50, 60, 70, 80 , 90 et 100%.
- On utilise aussi (en IR) : $\boxed{\textbf{F-mesure} = \frac{2 \times recall \times precision}{recall + precision} = \frac{2.TP}{2.TP + FP + FN}}$
- Une autre mesure utilisée :

$$\boxed{taux\ de\ TP \times (1\text{- }taux\ de\ FP) = \frac{TP \times TN}{(TP + FP) \times (FP + TN)}}$$

- Sans oublier le (bon) vieux **taux de succès** $= \frac{TP + TN}{TP + FP + TN + FN}$

Évaluation des schémas numériques

- Voir chapitre 4 des méthodes numériques
- Et les évaluations du chapitre 5 pour les numériques

Complément : Méthode Ensemble

- BUT : produire plusieurs modèles chacun partiellement compétent et les utiliser pour devenir *omnipotent*.

  ➙ Comme si on utilisait plusieurs experts de différents domaines.

  ➙ Comme dans un hôpital disposant de plusieurs spécialistes dans différentes maladie :

  ➙ l'ensemble (= l'hopital) est omnipotent.

- La méthode **Ensemble** suit le processus habituelle :

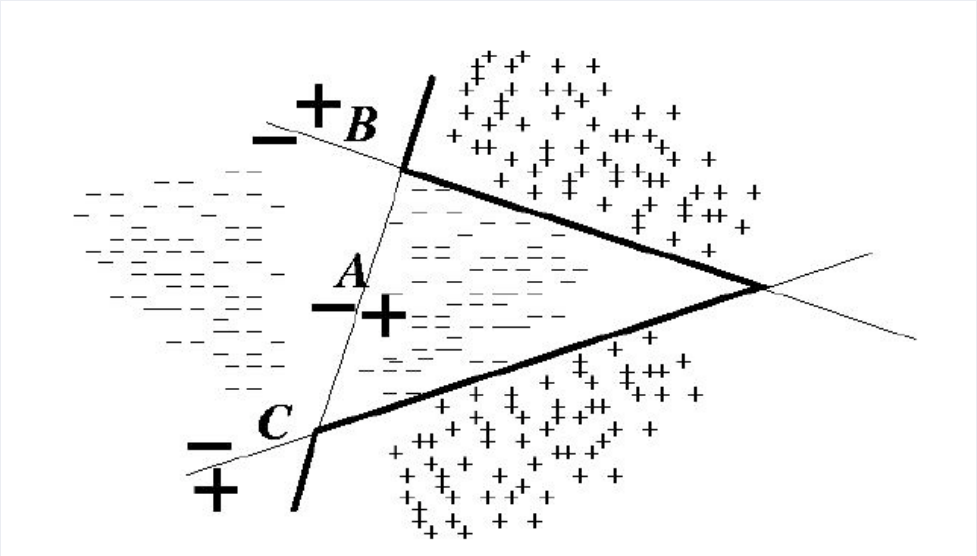
  ➙ **Préparation** (contrôle des données et échantillonage, partitionnement), Choix d'attributs, Application, Évaluation et Choix du **modèle EC**,..

  ➙ Puis un processus de choix de modèle utilisant des outil d'évaluation pour déterminer la spécificité, la sensibilité et la justesse (*accuracy*) ou la mauvaise classification (*misclassification*) du modèle final.

  ➙ et enfin la **décision**.

Complément : Méthode Ensemble (suite)

Méthode Ensemble : un exemple (Boosting & Bagging)



➙ Dans cette figure, un ensemble de 3 classifieurs linéaires (A,B,C) constitue conjointement le modèle.

➙ Les lignes en **gras** est l'**ensemble** qui classifie (ici bi-classes) un nouvel exemple en utilisant le vote majoritaire de A, B et C.

Complément : Méthode Ensemble (suite)

- Deux techniques sont utilisées dans la méthode ensemble :
  ➙ *Boosting* et *Bagging*.

Les données :

- les ensembles d'apprentissages sont différents
- ou il existe différentes distributions de pondérations sur un même ensemble d'apprentissage,
- voire, il existe différents sous ensembles d'attributs qui ont une forte corrélation avec les classes (la décision).

- La **décision finale** (la classe induite) peut être :
  ➙ Un vote majoritaire ou
  ➙ Une moyenne si les modèles produisent des probabilités de classement ou
  ➙ Une pondération des résultats de chaque modèle ou
  ➙ Une combinaison des modèles appris.

Complément : Méthode Ensemble (suite)

Bagging (ou *Bootstrap Aggregation*) : meilleur vote

- Échantillonne plusieurs *sous ensembles des données* d'apprentissage (avec remise)
- Procède à un apprentissage pour chaque sous ensemble (via des méthodes différentes)

- **Algorithme de principe de Bagging** :

Bagging(T, M) :
  Pour tout $m = 1, 2, ..., M$
    T_m = Sample_With_Replacement(T, | T |)
    $h_m = L_b(T_m)$ ←*appliquer la méthode L_b donnant le modèle h_m*
  Retourner $h_{fin}(x) = argmax_{y \in Y} \sum_{m=1}^{M} I(h_m(x) = y)$

- T est l'ensemble d'apprentissage de taille N
- M est le nombre de modèles à apprendre
- L_b est une méthode (algo) basique d'apprentissage (e.g. ID3 ou C45)
- h_i est un modèle basique (résultats) produit par L_b

- $I(A)$ renvoie 1 si $A = vrai$, 0 sinon.
  - Ici $I(h_m(x) = y) = 1$ si le modèle $h_m(x)$ classe correctement l'instance x dans sa vraie classe y (connue d'avance).

- Dans l'algo, *Sample_With_Replacement*(T, N) construit un ensemble d'apprentissage par N tirages aléatoires (avec remise) d'instances dans T.

Complément : Méthode Ensemble (suite)

• Bagging crée M modèles et produit la fonction $h(x)$ qui classifie toute nouvelle instance et renvoie sa classe y qui aura obtenu le maximum de vote des modèles $h_1, h_2, ..., h_M$.

• Sachant que les ensembles d'apprentissage T_m issus des tirages (dits *bootstrap*) sont différents, si les modèles appris sont différents tout en ayant des performances raisonnablement correctes, alors l'*Ensemble* produira des résultats meilleurs que chacun des modèles individuels.

• Il a été démontré qu'un *Ensemble* utilisant des *arbres de décision* donnent des résultats très intéressants.

☞ Voir la méthode *Random Forest*.

Complément : Méthode Ensemble (suite)

Boosting : exemple de la méthode *AdaBoost* (combinaison de modèles)

• *AdaBoost* génère une séquence de M modèles avec différentes pondérations de distribution sur le même ensemble E d'apprentissage.

Principe de l'algorithme AdaBoost :

• En entrée de cet algorithme, on a un ensemble E de N instances, une méthode d'apprentissage de base L_b
  ➙ Cherchons à construire M modèles d'apprentissage que nous souhaitons **combiner**.

• On construit une distribution D_1 sur l'ensemble d'apprentissage E. D_1 est en général uniforme à la 1e itération.
  ➙ Cette distribution assigne des poids (égaux pour la 1e itération) aux N exemples d'apprentissage.

• Pour $m = 1$, on construit un modèle de base h_1 à l'aide de la méthode L_b avec la distribution D_1 sur E.
  ➙ Avec le modèle h_1 obtenu, on calcule une erreur ε_1 sur le même ensemble d'apprentissage (E, D_1) ;
  ➙ Cette erreur est la somme des poids des instances mal classées par h_1.

• On doit obtenir l'erreur $\varepsilon_1 < 1/2$ (apprentissage *faible* mais mieux que pile-ou-face).

• Si cette condition n'est pas remplie, on arrête l'itération et on renvoie l'ensemble contenant le modèle précédemment calculé (si ce modèle n'existe pas, on change de méthode et on recommence).

• Si cette condition est satisfaite, on calcule une nouvelle distribution D_2 sur les instances de la manière suivante :
  ➙ Les instances qui sont correctement classées par h_1 voient leur poids multiplié par $\frac{1}{2(1-\varepsilon_1)}$
  ➙ Les instances mal classées par h_1 voient leur poids multipliés par $\frac{1}{2\varepsilon_1}$

• On remarque que les instances mal classées par h_1 voient leur poids réduits et celles bien classées voient leur poids augmenter. dans D_2.

• On passe ensuite à l'itération suivante en construisant h_2 en utilisant l'ens. d'appr. et la distribution D_2. . . .

• On construit M modèles de cette manière

Complément : Méthode Ensemble (suite)

• On note que le modèle de base suivant aura une erreur inférieure à 1/2 car les instances mal classées par le modèle précédent seront bien classées par le modèle suivant.

➙ De cette manière, *Boosting* force les modèles de base à corriger les mauvaises classifications par les modèles précédents.

• On construit M modèles de cette manière :

➙ La fonction renvoyée par AdaBoost affectera à toute nouvelle instance la classe qui aura le maximum de votes pondérés sur les M modèles

➙ Le poids de chaque modèle est $log(\frac{1-\varepsilon_m}{\varepsilon_m})$ qui est proportionnel à la justesse du modèle de base appliqué à l'ensemble d'apprentissage pondéré.

• *AdaBoost* a été au départ prévu pour les problèmes bi-classes mais elle peut être utilisée pour le cas général.

Complément : Méthode Ensemble (suite)

- **Algorithme AdaBoost :**

AdaBoost($\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}, L_b, M$)

Initialiser $D_1(n) = 1/N$ pour tout $n \in \{1, 2, .., N\}$

Pour tout $m = 1, 2, \ldots, M$

$h_m = L_b(\{(x_1, y_1), (x_2, y_2), \ldots, (x_N, y_N)\}, D_m)$

$$\varepsilon_m = \sum_{n : h_m(x_n) \neq y_n} D_m(n)$$

Si $\varepsilon_m =\geq 1/2$ alors $M = m - 1$ et stopper cette itération

Mettre à jour la distribution D_m par :

$$D_{m+1}(n) = D_m(n) \times \frac{1}{2(1 - \varepsilon_m)} \text{ si } h_m(x_n) = y_n$$

$$D_{m+1}(n) = D_m(n) \times \frac{1}{2\varepsilon_m} \text{ sinon}$$

Retourner $h_{fin}(x) = argmax_{y \in Y} \sum_{m=1}^{M} I(h_m(x) = y) log(\frac{1 - \varepsilon_m}{\varepsilon_m})$

Remarques sur les méta méthodes :

- Pour un cas bi-classes, le mieux-disant l'emporte (Vote si >2 classes)
- Échantillonnage des attributs : *Random Forest*

Table des matières