Data Mining 2 – Modélisation Statistique et Apprentissage
Master Statistique & Econométrie
UT1 Sciences Sociales, Toulouse
E-thème 3 **14 février 2018**
Enseignant: Gilles Cohen

Comparaisons et évaluations d'algorithmes d'apprentissage

1 Introduction

La performance du modèle $\mathcal{M}$ issu d'un algorithme d'apprentissage s'évalue par sa capacité de prédiction appelée erreur en généralisation. La mesure de cette performance est d'une grande importante puisque, d'une part, elle permet de procéder à la sélection d'un bon modèle dans une famille associée à l'algorithme d'apprentissage utilisé et, d'autre part, elle guide le choix de l'algorithme en comparant chacun des modèles optimisés à l'étape précédente.

2 Risque réel ou erreur de prédiction

Soint un ensemble d'apprentissage $S_n : \{(\mathbf{x}_i, y_i)_{i=1}^n\}$ constitué de n observations indépendantes et identiquement distribuées selon la distribution (i.i.d.) [1] $P(\mathbf{x}, y) = P(\mathbf{x})P(y|\mathbf{x})$. On souhaite estimer le modèle suivant

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

avec $\mathbb{E}(\epsilon) = 0, \mathrm{Var}(\epsilon) = \sigma^2$ et ϵ indépendant de $\mathbf{x}$. L'erreur de prévision ou risque associé au modèle est définie comme l'espérance de la fonction de perte :

$$R(h) = \mathbb{E}[\ell(y, h(\mathbf{x}))] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(y, h(\mathbf{x}))dP(\mathbf{x}, y) \tag{1}$$

où ℓ est une fonction *fonction de perte* qui mesure l'écart et le coût de l'écart entre la prédiction $h(\mathbf{x})$ du modèle et la valeur y donnée par l'ensemble

1. Considérant la distribution produit $P(\mathbf{x}, y) = P(y|\mathbf{x})P(\mathbf{x})$, l'échantillon de n exemples est tiré selon la mesure produit $P((\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)) = \prod_n P(\mathbf{x}_i, y_i)$.

d'apprentissage. Si y est quantitative, cette fonction de perte est généralement la perte quadratique : $\ell(h(\mathbf{x}), y) = \frac{1}{2}(y - h(\mathbf{x}))^2$. Si y est qualitative, $\ell(h(\mathbf{x}), y) = \mathbb{1}_{h(\mathbf{x}) \neq y}$ où $\mathbb{1}$ est la fonction indicatrice : $\mathbb{1}_P = 1$ si la proposition P est vraie. Cette fonction de perte peut être utilisée pour compter le nombre d'exemples mal classés.

2.1 Décomposition

On peut décomposer l'erreur de prédiction dans le cas quantitatif de la façon suivante :

$$\begin{aligned}
MSE(\hat{R}_n(x)) &= \mathbb{E}[(Y - \hat{R}_n(X))^2 | X = x] \\
&= \mathbb{E}[\mathbb{E}[(Y - \hat{R}_n(X))^2 | X = x, \hat{R}_n = \hat{r}] | X = x] \\
&= \mathbb{E}[\sigma_x^2 + (r(x) - \hat{R}_n(x))^2 | X = x] \\
&= \sigma_x^2 + \mathbb{E}[(r(x) - \hat{R}_n(x))^2 | X = x] \\
&= \sigma_x^2 + \mathbb{E}[(r(x) - \mathbb{E}[\hat{R}_n(x)] + \mathbb{E}[\hat{R}_n(x)] - \hat{R}_n(x))^2] \\
&= \sigma_x^2 + (r(x) - \mathbb{E}[\hat{R}_n(x)])^2 + Var(\hat{R}_n(x)) \\
&= \sigma^2 + \text{ Biais }^2 + \text{ Variance}
\end{aligned}$$

Très généralement, plus un modèle est complexe, plus il est flexible et peu s'ajuster aux données observées et donc plus le biais est réduit. En revanche, la partie variance augmente avec le nombre de paramètres à estimer et donc avec cette complexité. L'enjeu, pour minimiser le risque quadratique ainsi défini, est donc de rechercher un meilleur compromis entre biais et variance : accepter de biaiser l'estimation comme par exemple en régression ridge pour réduire plus favorablement la variance.

3 Mesures de performance

La façon la plus directe pour évaluer la performance des classifieurs est basée sur l'analyse de la matrice de confusion, qui est un résumé de la performance d'un classifieur. La matrice a deux dimensions. La première dimension correspond à la distribution des classes réelles de l'ensemble de test alors que la seconde dimension correspond à la distribution des classes prédites.

La figure **??** montre une matrice de confusion pour un problème à deux classes ayant pour valeur de classe positif et négatif, et la somme des valeurs des 4 cases donne la taille de l'ensemble de test (N). Il y a plusieurs mesures qui peuvent être déduites de la matrice de confusion. La mesure la plus

	Classe prédite	
Classe réelles	Vrai	Faux
Positif	Vrai Positif (VP)	Faux Négatif (FN)
Negatif	Faux Positif (FP)	Vrai Négatif (VN)

FIGURE 1 – Matrice de confusion pour un problème à deux classes. VP et VN représentent respectivement le nombre de vrais positifs et de vrais négatifs, c'est à dire, les cas positifs/négatifs reconnus comme tels par le classifieur. FP et FN représentent respectivement le nombre de cas positifs et négatifs mal classés par le classifieur.

simple est le taux d'erreurs, qui est défini comme le nombre de mauvaises classifications faites par le classifieur divisé par la taille de l'échantillon de test :

$$\text{Taux d'erreur} = \frac{|FN| + |FP|}{N}$$

De façon similaire, on peut aussi définir le taux de bien classés comme le nombre de prédictions correctes du classifieur divisé par la taille de l'échantillon test :

$$\text{Taux de bien classés} = \frac{|TP| + |TN|}{N}$$

On peut noter que la somme du taux d'erreurs et le taux de bien classés vaut toujours 1, et ces mesures sont symétriques, car elles ne font pas de distinction entre les exemples "positifs" et "négatifs". Deux mesures qui quantifient la performance d'un classifieur dans la prédiction d'exemples selon leur classe spécifique sont le taux de faux positifs et le taux de vrais positifs, qui quantifient respectivement le taux de faux positifs et de vrais positifs, dans leur classes réelles

$$\text{Taux de faux positifs} = \frac{|FP|}{|FP| + |TN|}$$

$$\text{Taux de vrai positifs} = \frac{|TP|}{|TP| + |FN|}$$

Le taux de vrai positifs est également appelé rappel ou sensibilité, et mesure la capacité du classifieur à détecter les exemples "positifs".

La précision mesure la capacité de prédiction du classifieur pour les positifs, c'est à dire, c'est le nombre de vrais positifs divisés par le nombre de positifs :

$$\text{Précsision} = \frac{|TP|}{|TP| + |FP|}$$

La spécificité, mesure la capacité de prédiction pour les négatifs et c'est le nombre de vrais négatifs divisés par le nombre de négatifs :

$$\text{Précsision} = \frac{|TN|}{|FP| + |TN|}$$

Il est intéressant de noter que la somme du taux de faux positifs et la spécificité donne toujours 1. De plus, plus grande est la spécificité plus faible est le nombre de faux positifs, mais c'est souvent associé à plus de faux négatifs.

4 Estimation du risque réel

4.1 Erreur d'apprentissage

Dans ce cas, l'estimation du risque réel $R(h_n)$ est simplement donnée par l'erreur obtenue sur l'ensemble d'apprentissage $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$

$$R_n(S) = \frac{1}{n} \sum_{i=1}^{n} \ell(h_n(\mathbf{x}_i), y_i)$$

Il s'agit d'un estimateur fortement biaisé puisque l'hypothèse choisie par l'algorithme d'apprentissage est la mieux adaptée à l'ensemble d'apprentissage et donc ses performances sur cet ensemble fournissent une vue optimiste de ses performances réelles.

4.2 Erreur de validation simple

Ici l'ensemble d'apprentissage S_n, de taille n, est divisé de façon aléatoire en un sous-ensemble d'apprentissage S_a de taille a, et un sous-ensemble de validation S_t de taille t $(S = S_a \cup S_t) \wedge S_a \cap S_t = \emptyset)$. L'algorithme d'apprentissage utilise l'ensemble S_a pour apprendre, alors que l'ensemble S_t sert

d'ensemble indépendant de test pour estimer l'erreur réelle du classifieur. L'estimateur de validation simple $R^t_{\text{VS}}(h_a)$ est :

$$R^t_{\text{VS}}(h_a) = \frac{1}{t}\sum_{i=1}^{t} \ell(h_a(\mathbf{x}_i, y_i)) \tag{2}$$

Habituellement on prend 2/3 des données pour l'apprentissage et on garde le reste pour la validation. Cet estimateur est non biaisé et sa variance diminue lorsque la taille de l'ensemble de validation augmente. Cette méthode n'est donc justifiable que lorsque le nombre de données est très important.

4.3 Validation croisée à k partitions

La validation croisée (VC) est une technique populaire pour estimer l'erreur en généralisation. Dans la validation croisée à k partitions l'ensemble d'apprentissage d'origine S est partitionné de façon aléatoire en k sous-ensemble S_j de taille approximativement identique. L'algorithme d'apprentissage est entrainé sur l'union des $(k-1)$ sous-ensembles ; le k-ième sous-ensemble restant est utilisé comme ensemble de test et mesure la performance en classification associée. Cette procédure est répétée sur tous les k ensembles de test possibles, et la moyenne de l'erreur de test donne une estimation de l'espérance de l'erreur en généralisation. L'erreur de la VC sur un ensemble d'apprentissage $S = \{(\mathbf{x}, y)\}_{i=1}^n$ est définie par

$$R^k_{\text{VC}}(S) = \frac{1}{k}\sum_{j=1}^{k}\left(\sum_{i=1}^{|S_j|} \ell(h_{S^{\setminus S_j}}(\mathbf{x}_i), y_i)\right) \tag{3}$$

où $S^{\setminus S_j}$ est l'ensemble de données obtenues en retirant le sous-ensemble S_j de l'ensemble S, $f_{S^{\setminus S_j}}$ est le classifieur correspondant et $|S_j|$ la cardinalité du sous-ensemble S_j.

4.4 Erreur Leave-One-Out

L'erreur leave-one-out (LOO) est un cas extrême de validation croisée dans lequel k est égal au nombre d'exemples. L'erreur LOO sur un ensemble d'apprentissage $S = \{(\mathbf{x}, y)\}_{i=1}^n$ est définie par

$$R_{\text{LOO}}(S) = \frac{1}{n}\sum_{i=1}^{n} \ell(h_{S^{\setminus i}}(\mathbf{x}_i), y_i) \tag{4}$$

où $h_S = \mathcal{A}(S)$ est le classifieur appris par l'algorithme $\mathcal{A}$ à partir de S, $S^{\setminus i} = S \setminus \{(\mathbf{x}_i, y_i)\}$ l'ensemble de données obtenue en retirant le i-ème exemple, et $f_{S^{\setminus i}}$ le classifieur correspondant.

Cette mesure compte le nombre d'exemples mal classés si on les exclut de l'apprentissage. Le théorème suivant montre que R_{LOO} est un estimateur presque non biaisé.

Théorème 4.1 *Soit P une distribution sur $\mathcal{X} \times \mathcal{Y}$, et S_n, S_{n-1} des ensembles de données de taille respective n et $n-1$ tirés i.i.d selon P. On note $R(f_{S_{n-1}})$ le risque espéré d'un estimateur obtenu à partir de l'ensemble S_{n-1} . Alors, pour tout algorithme, l'erreur LOO est presque non biaisée,*

$$\mathbb{E}_{S_{n-1}}[R(f_{S_{n-1}})] = \mathbb{E}_{S_n}[R_{LOO}(S_n)] \tag{5}$$

où $\mathbb{E}_{S_{n-1}}[R(f_{S_{n-1}})]$ est le le risque réel sur tous les choix possibles d'ensemble d'apprentissage de taille $n-1$.

Bien que ce théorème fournisse une bonne justification pour l'utilisation de la méthode LOO comme estimateur de l'erreur en généralisation, c'est néanmoins une méthode très coûteuse en temps puisqu'elle nécessite l'exécution de l'algorithme d'apprentissage n fois. Une stratégie est donc de borner supérieurement ou d'approximer cet estimateur par une quantité $\mathcal{B}$ ayant si possible une expression analytique.

4.5 Bootstrap

L'idée du bootstrap est d'approcher par simulation la distribution d'un estimateur lorsque la loi de l'échantillon est inconnue.
Pour cela, le principe de cette technique de rééchantillonnage consiste à substituer, à la distribution de probabilité inconnue P, dont est issu l'échantillon d'apprentissage, la distribution empirique P_n. Soit un ensemble d'apprentissage $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}$. L'idée de base est de créer des *ensembles bootstrap* en échantillonnant $p = |S|$ exemples avec remise à partir de S. On crée ainsi B ensembles bootstrap : $S_1, S_2, \ldots, S_B$. Ces ensembles ne sont pas mutuellement exclusifs, i.e. leur intersection n'est pas nulle. Afin d'estimer le risque réel d'un modèle une approche consiste à produire B hypothèses $h_{S_1}, \ldots, h_{S_B}$ à partir des B ensembles bootstrap et de regarder leur comportement respectif sur l'ensemble d'apprentissage d'origine S. Si $h_{S_b}(\mathbf{x}_i)$ est la valeur prédite pour $\mathbf{x}_i$, à partir de l'hypothèse apprise sur l'ensemble

bootstrap b, l'estimation du risque réel s'écrit :

$$R_{\text{BS}}(S) = \frac{1}{B}\sum_{b=1}^{B}\frac{1}{n}\sum_{i=1}^{n}\ell(h_{S_b}(\mathbf{x}_i), y_i) \tag{6}$$

Cette estimation du risque réel est généralement biaisée par optimisme parce que les mêmes exemples $(\mathbf{x}_i, y_i)$ peuvent se retrouver à la fois dans l'estimation de l'hypothèse et dans celle de son erreur. En effet, un exemple test $\mathbf{x}_i$ est inclu dans l'ensemble bootstrap S_b avec une probabilité $1-\left(1-\frac{1}{n}\right)^n$ ce qui donne approximativement 0.632 pour de grandes valeurs de n. En s'inspirant de la validation croisée, une meilleure estimation peut être obtenue. Pour chaque exemple, on ne garde que la prédiction à partir de l'ensemble bootstrap qui ne contient pas l'exemple. Ce nouvel estimateur du risque réel, appelé bootstrap leave-one-out, est défini par :

$$R_{\text{BS-LOO}}(S) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C^{\setminus i}|}\sum_{b\in C^{\setminus i}}\ell(h_{S_b}(\mathbf{x}_i), y_i) \tag{7}$$

Ici $C^{\setminus i}$ est l'ensemble des indices b des échantillons bootstrap qui ne contiennent pas l'exemple i, et $|C^{\setminus i}|$ est le nombre de ces échantillons. Lorsque l'on calcule R_{boot} on devra soit prendre B suffisamment grand pour s'assurer que tous les $|C^{\setminus i}|$ sont plus grands que 0, soit ne pas prendre en compte dans Eq. (**??**) les termes correspondant à des $|C^{\setminus i}|$ nuls. Le bootstrap R_{oob} résoud le problème d'un biais optimiste mais conserve celui du biais de la taille de l'ensemble d'apprentissage de la validation croisée. Pour y remédier, certains auteurs ont proposé la correction suivante :

$$R_{\text{BS}-.632}(S) = 0.368.R_n(S) + 0.632R_{\text{BS-LOO}} \tag{8}$$

appelée boostrap 0.632.

4.6 Aire sous la courbe ROC

La courbe ROC (Receiver Operating Characteristic) peut être utilisée pour analyser la relation entre le taux de FN et le taux de FP (ou le taux de VN et le taux de VP) pour un classifieur. Une courbe ROC trace la sensibilité versus 1−spécificité pour différents seuils de sortie du classifieur. Sur la base de la courbe ROC, on peut décider de la quantité de faux positives (respectivement de faux négatifs) que l'on est prêt à tolérer et régler ainsi le seuil du classifieur pour s'adapter au mieux à une application donnée. Une affectation aléatoire des classes aux éléments d'un ensemble de données

produira une courbe ROC de la forme d'une ligne diagonale de point de départ $(0, 0)$ et de point d'arrivée $(1, 1)$.
Il a été démontré que l'aire sous une courbe ROC empirique correspond à la probabilité qu'un exemple appartenant à la classe des positifs obtienne un résultat plus élevé qu'un exemple de la classe des négatifs. Posons $\mathbf{x}^+$ et $\mathbf{x}^-$, les variables aléatoires représentant le résultat de la classification pour un exemple positif et un exemple négatif respectivement. En termes mathématiques, l'aire sous la courbe ROC s'écrit $\mathcal{A}_{\text{AUC}} = \mathbb{P}(\mathbf{x}^+ > \mathbf{x}^-)$, si $\mathbf{x}^+$ et $\mathbf{x}^-$ sont continues et satisfont $\mathbb{P}(\mathbf{x}^+ = \mathbf{x}^-) = 0$. L'aire sous la courbe est définie comme étant $\mathcal{A}_{\text{AUC}} = \mathbb{E}[g(\mathbf{x}^+, \mathbf{x}^-)]$ où

$$g(\mathbf{x}^+, \mathbf{x}^-) = \begin{cases} 1, & \mathbf{x}^+ > \mathbf{x}^-, \\ 0.5, & \mathbf{x}^+ = \mathbf{x}^-, \\ 1, & \mathbf{x}^+ < \mathbf{x}^- \end{cases} \tag{9}$$

Soit n_+ le nombre d'éléments de la classe des positifs et n_- celui de la classe des négatifs. La procédure pour calculer cette aire consiste à effectuer toutes les $n_+ \times n_-$ comparaisons possibles entre les exemples des deux classes en attribuant le score définit par g à chaque comparaison. L'estimation de l'aire sous la courbe est obtenue par

$$\hat{\mathcal{A}}_{\text{AUC}} = \frac{1}{n_+ n_-} \sum_{j=1}^{n_+} \sum_{k=1}^{n_-} g(\mathbf{x}_j^+ - \mathbf{x}_k^-)$$

Ce qui correspond au test de Mann-Whitney normalisé par $1/n_+n_-$ C'est un estimateur non biaisé de l'aire sous la courbe ROC.

5 Test d'hypothèses

Au lieu d'estimer explicitement certains paramètres, dans certaines applications on peut vouloir utiliser l'échantillon pour tester certaines hypothèses particulières relatives aux paramètres. Par exemple, au lieu de l'estimation de la moyenne, on peut vouloir tester si la moyenne est inférieure à 0.02. Si l'échantillon aléatoire est compatible avec l'hypothèse considérée on n'arrive pas à rejeter l'hypothèse, sinon, nous disons qu'elle est rejetée. Dans le test d'hypothèses, l'approche est la suivante. On définit une statistique qui obéit à une certaine distribution, si l'hypothèse est correcte. Si la statistique calculée à partir de l'échantillon a une probabilité très faible d'être issue de cette distribution, alors on rejette l'hypothèse, sinon, on ne peut pas la rejeter.

	Décision	
Réalité	Ne pas rejeter H_0	Rejeter H_0
H_0 bonne	Correct	Erreur type I
H_0 mauvaise	Erreur type II	Correct (puissance)

FIGURE 2 – Type d'erreur I, type d'erreur II, et puissance d'un test

.

On a un échantillon issue d'une distribution normale de moyenne inconnue μ et variance inconnue σ^2, et l'on veut tester une hypothèse spécifique au sujet de μ, comme par exemple si elle est éagle à une constante μ_0. On note cette hypothèse H_0 et on la nomme *hypothèse nulle* $H_0 : \mu = \mu_0$ par rapport à une hypothèse alternative $H_1 : \mu \neq \mu_0$. $\hat{\mu}$ est l'estimateur ponctuel de μ, et il est raisonnable de rejeter H_0 si $\hat{\mu}$ est trop éloignée de μ. C'est là où l'estimation de l'intervalle est utilisée. On ne rejette pas l'hypothèse à un niveau de significativité α si μ_0 se trouve dans l'intervalle de confiance $100(1-\alpha)$ %, c'est à dire si

$$\frac{\sqrt{N}(\hat{\mu} - \mu_0)}{\sigma} \in (-z_{\alpha/2}, z_{\alpha/2})$$

On rejette l'hypothèse si on se trouve en dehors de quel coté que ce soit. C'est un test bilatéral. Si l'on rejette lorsque l'hypothèse est correcte, c'est un type d'erreur I et donc la valeur α fixée avant le test indique la quantité d'erreur de type I que l'on tolère, des valeurs typiques de α sont $\alpha = 0.1, 0.05, 0.01$. L'erreur de type II signifie que l'on ne rejette pas l'hypothèse nulle lorsque la moyenne μ n'est pas égale à μ_0. La probabilité que H_0 ne soit pas rejetée lorsque la vraie moyenne est μ est une fonction de μ et est donnée par

$$\beta(\mu) \quad = \quad P_\mu\{-z_{\alpha/2} \leq \frac{\hat{\mu} - \mu_0}{\sigma/\sqrt{N}} \leq z_{\alpha/2}\}$$

$1 - \beta(\mu)$ est appelée la fonction de puissance du test et est égale à la probabilité de rejet lorsque μ est la vraie valeur. La probabilité de l'erreur type II croit au fur et à mesure que μ et μ_0 se rapproche et on peut calculer la taille d'un échantillon dont on a besoin être en mesure de détecter une différence $\delta = |\mu - \mu_0|$ avec une puissance suffisante.

6 Evaluation de la performance d'un algorithme de classification

Maintenant que l'on a passé en revue les tests d'hypothèses, on va voir comment il sont utilisés dans les tests des taux d'erreur.

6.1 Test binomial

On commence avec le cas où on a un seul ensemble d'apprentissage S_T et un seul ensemble de validation S_V. On apprend le classifieur sur S_T et on teste sur S_V. On note par p la probabilité que le classifieur produise une erreur de classification. On ne connait pas p ; et l'on voudrait l'estimer. Pour un exemple d'index t de l'ensemble de validation S_V, soit x_t qui désigne la justesse de la décision du classifieur ; x_t est une variable aléatoire de Bernoulli qui prend la valeur 1 lorsque le classifieur commet une erreur et 0 lorsqu'il n'en commet pas. La variable aléatoire binomiale X désigne le nombre total d'erreurs :

$$X = \sum_{i=1}^{N} x_i$$

On veut tester si la probabilité d'erreurs p est inférieure ou égale à une valeur p_0 on a :

$$H_0 : p \leq p_0 \text{ vs. } H_1 : p > p_0$$

Si la probabilité d'erreurs est p, la probabilité que le classifieur commette j erreurs sur N est

$$P(X = j) = \left(\begin{array}{c} N \\ j \end{array} \right) p^j (1-p)^{N-j}$$

Il est raisonnable de rejeter $p \leq p_0$ si dans un tel cas, la probabilité que l'on ait $X = e$ erreurs ou plus est trés peu probable. Autrement dit, le test binomial rejette l'hypothèse, si

$$P(X \geq e) = \sum_{\mathbf{x}=e}^{N} \left(\begin{array}{c} N \\ \mathbf{x} \end{array} \right) p_0^{\mathbf{x}} (1-p_0)^{N-\mathbf{x}} < \alpha$$

où α est la significativité, par exemple, 0.05.

6.2 t-test

Le précédent test utilise un seul ensemble de validation. Si l'on lance l'algorithme K fois, sur K paires d'ensembles d'apprentissage/validation, on obtient K pourcentages d'erreurs, $p_i, i = 1, \ldots, K$ sur les K ensembles de validation. Soit $x_t^i = 1$ si le classifieur appris sur S_T^i produit une erreur de classification sur l'exemple t de S_V^i ; $x_t^i = 0$ autrement. Alors

$$p_i = \frac{\sum_{t=1}^{N} x_t^i}{N}$$

Sachant que

$$\hat{\mu} = \frac{\sum_{i=1}^{K} p_i}{K},\ S^2 = \frac{\sum_{i=1}^{K} (p_i - \hat{\mu})^2}{K-1}$$

on sait que l'on a

$$\frac{K(\hat{\mu} - p_0)}{S} \sim t_{K-1}$$

et le t-test rejette l'hypothèse nulle que l'algorithme de classification a p_0 ou moins de pourcentage d'erreurs à un niveau de significativité α si cette valeur est plus grande que $t_{\alpha,K-1}$. En général, on prend comme valeur de K 10 ou 30. $t_{0.05,0} = 1.83$ et $t_{0.05,29} = 1.70$.

7 Comparaison de deux algorithmes de classifications

Soient deux algorithmes de classification, on veut comparer et tester si ils construisent deux classifeurs qui ont le même taux d'erreur.

7.1 Test de McNemar

Soient un ensemble d'apprentissage et un ensemble de validation, on utilise deux algorithmes pour apprendre à partir de l'esemble d'apprentissage deux classifieurs et on les teste sur l'ensemble de validation puis on calcule leurs erreurs. Une table de contingence, comme illustrée par la figure **??**, est une matrice d'entiers naturels représentant des décomptes ou des fréquences :

Sous l'hypothèse nulle que les algorithmes de classification aient le même taux d'erreurs, on s'attend à ce que $e_{01} = e_{10}$ et que $(e_{01} + e_{10}/2)$. On a la statistique du chi-carré à un degré de liberté

$$\frac{(|e_{01} - e_{10}| - 1)^2}{e_{01} + e_{10}} \sim \chi_1^2$$

e_{00} : Nombre d'exemples mal classés par les deux	e_{01} : Nombre d'exemples mal classé par 1 mais pas par 2
e_{10} : Nombre d'exemples mal classés par 2 mais pas par 1	e_{11} : Nombre d'exemples bien classés les deux

FIGURE 3 – Table de contingence

et le test de McNemar rejette l'hypothèse que les deux algorithmes de classification aient le même taux d'erruers avec un niveu de significativité α si cette valeur est plus grande que $\chi^2_{\alpha,1}$. Pour $\alpha = 0.05$, $\chi^2_{0.05,1} = 3.84$.

7.2 T-test apparié k- validation croisée

On utilise k-validation croisée afin d'obtenir k paires d'ensemble d'apprentissage/validation. On utilise deux algorithmes de classification pour apprendre sur les ensembles d'apprentissage $S^i_T, i = 1, \ldots, k$ et on teste sur les ensmbles de validation S^i_V. Les pourcentages d'erruers des classifieurs surl les ensembles de validations son enregistrés comme p^i_1 et p^2_i. Si les deux algorithmes de classifications ont le même taux d'erreurs, alors on s'attend à ce qu'ils aient la même moyenne, ou ce qui est équivalent que la différence de leur moyenne est 0. La différence sur le taux d'erreurs sur la partition i est $p^i = p^i_1 - p^i_2$. C'est un test apparié ; c'est à dire que pour chaque i, les deux algorithmes ont les mêmes ensembles d'apprentissage et de test. Lorsque que les k fois ont été réalisées, on a une distribution de p^i contenant k points. Ssachant que p^i_1 et p^i_2 suivent toutes les deux (approximativement) une distribution normale, leur différence p^i est également une distribution normale. L'hypothèse nulle est que la distribution a une moyenne égale à 0 :

$$H_0 : \mu = 0 \text{ vs. } H_1 : \mu \neq 0$$

on définit

$$\hat{\mu} = \frac{\sum_{i=1}^{k} p^i}{k}, S^2 = \frac{\sum_{i=1}^{k} (p^i - \hat{\mu})^2}{k-1}$$

Sous l'hypothèse nulle que $\mu = 0$, on a une statistique qui est distribuée selon une loi de Student $k-1$ degrés de libertés :

$$\frac{\sqrt{k}(\hat{\mu} - 0)}{S} = \frac{\sqrt{k}.\hat{\mu}}{S} \sim t_{k-1}$$

donc le t-test apparié de la k validation croisée rejette l'hypothèse que les deux algorithmes de classification aient le même taux d'erreur à un niveu de significativité α si cette valeur se trouve en dehors de l'intervalle

$(-t_{\alpha/2,k-1}, t_{\alpha/2,k-1})$. Pour $\alpha = 0.05$ on a $-t_{0.025,9} = 2.26$ et $t_{0.025,9} = 2.05$ Si l'on veut tester si le premier algorithme commet moins d'erreurs que le second, on a alors besoin d'une hypothèse unilatéral et on utilise un test :

$$H_0 : \mu \geq 0 \text{ vs. } H_1 : \mu < 0$$

Si le test rejette l'affirmation selon laquelle le premier a beaucoup moins d'erreurs est confirmée.