classification_multiple

May 16, 2022

1 Classification multiple

Explorations autour d'un problème de classification multiple.

[1]:
```
from jyquickhelper import add_notebook_menu
add_notebook_menu()
```

[1]:
```
<IPython.core.display.HTML object>
```

1.1 Début de l'histoire

$\mathbb{1}_{y_i}$

1.1.1 Confusions

Un des premiers réflexes après avoir appris une classification multi-classe est de regarder la matrice de confusion. Certaines classes sont difficiles à classer, d'autres non. Je me demandais s'il existait un moyen de déterminer cela sans apprendre un classifieur. On souhaite apprendre la classification des points (X_i, y_i), X_i est un vecteur, y_i la classe attendue. Si $\hat{y_i}$ est la classe prédite, l'erreur de classification est :

$$E = \sum_i \mathbb{1}_{y_i \neq \hat{y_i}}$$

On note $c_{ij} = \mathbb{1}_{y_i=j}$ et $\hat{c_{ij}} = \mathbb{1}_{\hat{y_i}=j}$. On note le vecteur $C_j = (c_{ij})_i$ et $\hat{C_j} = (\hat{c_{ij}})_i$. On peut réécrire l'erreur comme :

$$E = \sum_{ij} \mathbb{1}_{y_i=j} \mathbb{1}_{\hat{y_i} \neq j} = \sum_{ij} \mathbb{1}_{y_i=j}(1 - \mathbb{1}_{\hat{y_i}=j}) = \sum_{ij} c_{ij}(1 - \hat{c_{ij}}) = \sum_j < C_j, 1 - \hat{C_j} >$$

C'est aussi égal à :

$$E = \sum_{k \neq j} < C_j, \hat{C_k} >$$

Et $< C_j, \hat{C_k} >$ correspond au nombre d'erreurs de confusion : le nombre d'éléments de la classe j classés dans la classe k. $< C_j, \hat{C_k} >$ est le nombre d'éléments correctement classés dans la classe j. On peut montrer que

$$\sum_{k,j} < C_j, \hat{C_k} >= N$$

où N est le nombre d'observations.

1.1.2 Clustering

Et si nous introduisions un clustering intermédiaire. On construit Q cluster, q_i est le cluster du point X_i et on note $d_{il} = \mathbb{1}_{q_i=l}$ et le vecteur $D_l = (d_{il})_i$.

$$E = \sum_{k \neq j} < C_j, \hat{C_k} >$$

On note $X.Y$ le produit terme à terme de deux vecteurs.

$$E = \sum_{k \neq j,l} < C_j.D_l, \hat{C_k} >= \sum_{k \neq j,l} < C_j.D_l, \hat{C_k}.D_l >$$

Le nombre d'erreurs est la somme des erreurs faites sur chaque cluster. Supposons maintenant qu'un classifieur retourne une réponse constante sur chacun des clusters, on choisit la classe plus représentée. Ca ressemble beaucoup à un classifieur bayésien. On note $f(l)$ cette classe la plus représentée. Elle vérifie :

$$f(l) = \arg\max_j < C_j, D_l >$$

Cela signifie que $\hat{c_{ij}} = \sum_l \mathbb{1}_{j=f(l)} d_{il}$. Si on note $l(i)$ le cluster associé à i. On continue : $\hat{c_{ij}} = \mathbb{1}_{j=f(l(i))}$. On définit l'erreur $e(l)$ l'erreur de classification faite sur chaque cluster l :

$$e(l) = \sum_i d_{il} \sum_j c_{ij}(1 - \mathbb{1}_{j=f(l)}) = \sum_i d_{il} \left(\sum_j c_{ij} - \sum_j c_{ij} \mathbb{1}_{j=f(l)} \right) = \sum_i d_{il} \left(1 - c_{i,f(l)}\right) = \sum_i d_{il} - \sum_i d_{il} c_{i,f(l)}$$

Pour résumer, l'erreur est le nombre d'éléments moins le nombre d'éléments dans la classe majoritaire du cluster. Si le nombre de clusters Q devient supérieur ou égal au nombre d'observations, cette erreur devient nulle.

1.2 Mise en pratique

L'idée est de voir comment évolue cette erreur de classification naïve en fonction du nombre de clusters. La différence par rapport à un classifieur est qu'on sait comment sont fabriqués les clusters et qu'on peut imaginer les classes comme un assemblage de clusters d'une forme connue.

[2]: