

Projet Crowdsourcing

CÔME, SENE

Présentation de projet de Master 2

3 Janvier 2023



Introduction

Contexte

- Utilisation de l'Algorithme EM pour connaître le vrai label d'une image donnée ou déterminer la véritable pathologie d'un patient.
- Type de données : 60000 images 32×32 séparées en 10 classes
- Problématique : La labélisation d'une image donnée ou le diagnostic d'un patient.
- Approche classique : affichage de la matrice de confusion à l'aide de l'algorithme EM.



- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques
- 6 Expériences numériques
- 7 Conclusion



Notations

Notations

On note $\forall i \in \{1, 2, \dots, I\}, \forall j, l \in \{1, 2, \dots, J\},$
 $\forall k \in \{1, 2, \dots, K\} :$

- π_{lj}^k : la probabilité que le médecin k donne la réponse j sachant que la vraie réponse est l .
- T_{ij} : une variable de réponse associée au patient i définie par $T_{iq} = 1$ si q est la vraie réponse et $T_{iq} = 0$ si $j \neq q$.
- p_j : la prévalence de la classe j ou la fréquence empirique.



- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques
- 6 Expériences numériques
- 7 Conclusion



Cas 1

1 medecin 1 patient

Soit X à valeur dans $\{1, 2, \dots, M\}$, une variable aléatoire indiquant la maladie du patient.

Soit Y à valeur dans $\{1, 2, \dots, J\}$, une variable aléatoire correspondant à la maladie du patient indiquée par le médecin.

$$Y|X = x \sim \text{Multinomiale} \left[(\pi_{xl}^Y)_l, n \right], \text{ avec } l \in \{1, 2, \dots, J\}.$$



Cas 1

1 médecin 1 patient

Sa fonction de masse est donnée par :

$$\mathbb{P} \left(n_{i1}^k, \dots, n_{iJ}^k \right) = \frac{\left[\sum_{j=1}^J n_{ij}^k \right]!}{\prod_{j=1}^J n_{ij}^k!} \prod_{j=1}^J \left(\pi_{lj}^k \right)^{n_{ij}^k} \propto \prod_{j=1}^J \left(\pi_{lj}^k \right)^{n_{ij}^k}.$$

Ainsi la vraisemblance est :

$$\propto \prod_{j=1}^J \left(\pi_{lj}^k \right)^{n_{ij}^k}$$



Cas 2

K medecins et I patients

Soit $X = (X_1, \dots, X_I)$ le vecteur aléatoire indiquant la maladie des I patients.

Soit $Y = (Y_1, \dots, Y_K)$ le vecteur aléatoire indiquant la maladie des K patients.

Si le médecin répond une fois à la question du patient, on a :

$$Y_i^k | X_i = x_i \sim \text{Multinomiale} \left[(\pi_{x_i j}^k), 1 \right],$$

avec $j \in \{1, 2, \dots, J\}$ et $i \in \{1, 2, \dots, I\}$



Cas 2

K medecins et I patients

Étant donné que les (Y_i^k) sont indépendants

$\forall k \in \{1, 2, \dots, K\}$ et $i \in \{1, 2, \dots, I\}$, Donc :

$$\mathbb{P}(n_{i1}^k, \dots, n_{iJ}^k) = \frac{[\sum_{j=1}^J n_{ij}^k]!}{\prod_{j=1}^J n_{ij}^k!} \prod_{j=1}^J (\pi_{x_{ij}}^k)^{n_{ij}^k} \propto \prod_{k=1}^K \prod_{j=1}^J (\pi_{x_{ij}}^k)^{n_{ij}^k}.$$

Par conséquent la vraisemblance est :

$$\propto \prod_{k=1}^K \prod_{j=1}^J (\pi_{x_{ij}}^k)^{n_{ij}^k}$$



Cas 2

Toutes les données

Si on se base sur toutes les données c'est-à-dire les réponses de tous les médecins et les questions de tous les patients, on a :

$$\mathbb{P} \left((\cap Y_i^k) \mid \cap (X_i = x_i) \right) \mathbb{P} (\cap (X_i = x_i))$$

=

$$\mathbb{P} \left((\cap Y_i^k) \cap (\cap (X_i = x_i)) \right).$$

Par indépendance des (Y_i^k) et (X_i) ,



Cas 2

Toutes les données

$$\begin{aligned}
 \mathbb{P} \left((\cap Y_i^k) \cap (\cap (X_i = x_i)) \right) &= \prod_{i=1}^I \mathbb{P} \left(\cap Y_i^k \mid \cap (X_i = x_i) \right) \mathbb{P}(X_i = x_i) \\
 &= \prod_{i=1}^I \left(\mathbb{P}(X_i = x_i) \prod_{k=1}^K \mathbb{P}(Y_i^k \mid \cap (X_i = x_i)) \right) \\
 &= \prod_{i=1}^I \left(p_{x_i} \prod_{k=1}^K \mathbb{P}(Y_i^k \mid (X_i = x_i)) \right)
 \end{aligned}$$



Cas 2

Toutes les données

D'où

$$\mathbb{P}\left(\left(\cap \cap Y_i^k\right) \cap \left(\cap (X_i = x_i)\right)\right) \propto \prod_{i=1}^I \prod_{l=1}^J \left[p_{x_i} \prod_{k=1}^K \prod_{j=1}^J \left(\pi_{ij}^k \right)^{n_{ij}^k} \right]^{T_{ij}}.$$

Par conséquent la vraisemblance de toutes les données est :

$$\propto \prod_{i=1}^I \prod_{l=1}^J \left[p_{x_i} \prod_{k=1}^K \prod_{j=1}^J \left(\pi_{ij}^k \right)^{n_{ij}^k} \right]^{T_{ij}}$$



- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance**
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques
- 6 Expériences numériques
- 7 Conclusion



Les estimateurs

Expressions

Si on suppose que les T_{ij} sont connus et que en pratique nous avons les n_{ij}^k , on a les estimateurs du maximum de vraisemblance suivants :

$$\hat{\pi}_{jl}^k = \frac{\sum_{i=1}^I T_{ij} n_{il}^k}{\sum_{l=1}^J \sum_{i=1}^I T_{ij} n_{il}^k}.$$

$$\hat{p}_j = \frac{\sum_{i=1}^I T_{ij}}{I}.$$



- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques
- 6 Expériences numériques
- 7 Conclusion



cifar10h

Base de données composée de 10 classes d'images :

- classe 0 : airplane
- classe 1 : automobile
- classe 2 : bird
- classe 3 : cat
- classe 4 deer
- classe 5 : dog
- classe 6 : frog
- classe 7 : horse
- classe 8 : ship
- classe 9 : truck



cifar10h

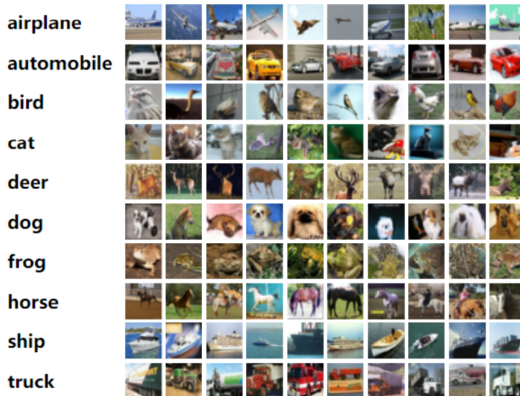


Figure – Un extrait de la base de données

- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques**
- 6 Expériences numériques
- 7 Conclusion



La fonction *CreateSubDf*

Son objectif

Créer un sous dataframe dans lequel seront stockées uniquement les données utiles :

Ses arguments

- ***df*** : le dataframe des données (Il s'agit du dataframe d'origine)
- ***an_id*** : entier (compris entre 0 et 2570) désignant l'identifiant de l'annotateur

Ce qu'elle retourne

Un sous dataframe composé de deux colonnes :

- ***La colonne 1*** : contient les labels choisis par un annotateur spécifique
- ***La colonne 2*** : contient les vrais labels associés aux images



La fonction *custom_confusion_matrix*

Son objectif

Améliorer l'esthétique de la représentation graphique de la matrice de confusion

Ses arguments

- ***cm*** : un 2D numpy array jouant le rôle de la matrice de confusion
- ***classes*** : la liste stockant les noms de chaque classe
- ***normalize*** : (Booléen) : si True la matrice sera normalisée, et si False elle ne le sera pas
- ***title*** : le titre du graphique
- ***cmap*** : la palette de couleur du graphique



La fonction `plot_confusion_matrix`

Son objectif

Calculer la matrice de confusion et afficher sa représentation graphique

Ses arguments

- **`y_true`** : un numpy array dans lequel sont stockés les vrais labels
- **`y_predict`** : un numpy array dans lequel sont stockés les labels choisis par l'annotateur
- **`normalize(Booléen)`** : si True la matrice sera normalisée, et si False elle ne le sera pas
- **`class_names`** : la liste stockant les noms de chaque classe

Ce qu'elle retourne

retourne le graphique de la matrice de confusion



La fonction *DawidSkenelID*

Son objectif

Permet d'initialiser le modèle de David Skene

Ses arguments

- ***ydims*** : un couple sous la forme (nombre de classes, nombre d'annotateurs)
- ***max_iter*** : Le nombre d'iterations souhaité de l'algorithme EM
- ***predict_tol*** : un seuil de tolérance pour la prédiction

Ce qu'elle retourne

retourne le graphique de la matrice de confusion



La fonction *fit*

Son objectif

Permet d'alimenter le modèle

Ses arguments

- ***U*** : les données de dim = (N,K) avec N lignes et K annotateurs (dans notre cas $k = 1$)
- ***priors*** : les probabilités du prior pour π_z et $\psi_{u,z}^k = p(u_k = u | Z = z)$
- ***starts*** : tuple contenant les matrices (π et ψ) des paramètres initiaux de l'algorithme EM



La fonction `plot_cm_d`

Son objectif

afficher la représentation graphique de la matrice de confusion

Ses arguments

- **U** : les données de dim = (N,K) avec N lignes et K annotateur (dans notre cas $k = 1$)
- **priors** : les probabilités du prior pour π_z et $\psi_{u,z}^k = p(u_k = u | Z = z)$
- **starts** : tuple contenant les matrices (π et ψ) des paramètres initiaux de l'algorithme EM

Ce qu'elle retourne

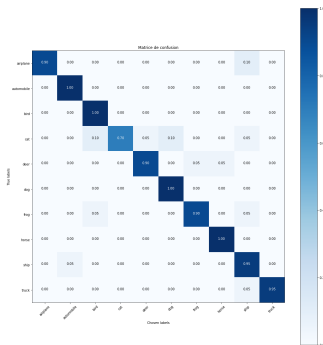
le graphique de la matrice de confusion estimée à l'aide de l'algorithme EM



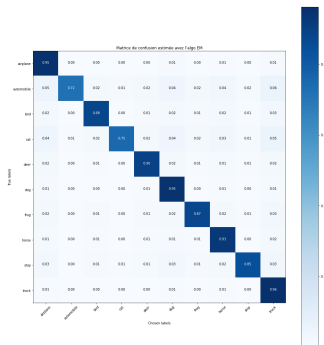
- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques
- 6 Expériences numériques**
- 7 Conclusion



Expériences numériques



(a) Matrice de confusion associée aux labels prédits par un annotateur



(b) Matrice de confusion estimée à partir de l'algorithme EM



- 1 Notations
- 2 Estimation de la vraisemblance
- 3 Estimation du maximum de vraisemblance
- 4 Le jeu de données
- 5 Fonctions utiles pour réaliser les expériences numériques
- 6 Expériences numériques
- 7 Conclusion



Conclusion

Conclusion

D'après nos expériences numériques, nous constatons que :

- l'algorithme EM prédit bien nos images c'est-à-dire qu'il fait une bonne labélisation. En revanche, sa matrice de confusion n'est pas forcément meilleure que celle des annotateurs.
- Cependant, malgré que la matrice de confusion des annotateurs ne montre pas beaucoup d'erreurs (presque diagonale), elle reste tout de même proche à celle de l'algorithme EM en terme de prédiction.

