ECOLE NATIONALE DES PONTS ET CHAUSSEES

Deuxième Semestre

COURS DE MACHINE LEARNING

MALAP

Rapport de Projet

Learning from crowds

Réalisé par: REIZINE Charles, MATHIEU Emile et PESNEAU Thomas

Sommaire

Liste des figures

Résumé

L'apprentissage supervisé pour des données avec plusieurs annotateurs est une problématique dont l'intérêt croit dans le milieu de l'apprentissage automatique. En effet, avec la popularité croissante de plateformes de crowdsourcing (tel Amazon Mechanical Turk), des quantités de plus en plus importante de données avec plusieurs annotateurs sont générées. Diverses approches ont ainsi été proposées afin de modéliser l'expertise des annotateurs qui peut varier grandement suivant l'annotateur mais aussi en fonction des données.

1 Introduction

1.1 Introduction

La facilité avec laquelle toute donnée peut être partagée, organisée et traitée par un nombre important d'entités utilisant des infrastructures de communications standards (comme l'Internet ou son concurrent le minitel) créé de nombreux problèmes et opportunités intéressants pour le domaine de l'apprentissage automatique. Ainsi, les connaissances de ces différentes entités, et en particulier des personnes, peuvent maintenant être facilement collectées et agrégées d'une manière complètement distribuée. Cependant, combiner des connaissances provenant de différentes sources est loin d'être un problème résolu.

Traditionnellement, l'apprentissage supervisé s'appuie sur l'expert d'un domaine qui joue le rôle de professeur en fournissant la supervisation nécessaire. La situation la plus commune étant celle où les annotations de cet expert sont utilisées comme labels dans des problèmes de classification. Cependant, avec l'avènement du phénomène de crowdsourcing([1]), des services tels que Amazon Mechanical Turk permettent de récupérer à bon marché et rapidement des quantités importantes de données labelées par de nombreux annotateurs. La qualité des annotateurs n'étant ici pas certifiée, il sera demandé à différents annotateur de labeliser un même échantillon. La particularité du crowds learning réside donc dans la pluralité des labels pour un même échantillon. Dans l'apprentissage supervisé classique, plus de données amènent, sous certaines hypothèses, à un classifieur plus précis. Comment ici plus d'annotateurs peut-il aussi se traduire en un classifieur plus précis ? Et comment les connaissances de chaque annotateur peuvent-elles être utilisées au mieux ?

La disponibilité croissante du nombre d'annotateurs n'est pas la seule motivation pour les méthodes d'apprentissages à plusieurs annotateurs. En effet, dans de nombreux domaines il est parfois impossible ou très coûteux d'obtenir le *ground truth* label. Par exemple, pour la détection de cancer par imagerie médicale, seule une biopsie permet de réellement savoir si une région du corps est réellement cancéreuse ou non; cette opération est coûteuse et peut être dangereuse. Par ailleurs, de nombreux travaux d'annotations sont subjectifs par nature et il n'existe donc pas clairement de label correct.

Une façon intuitive et naïve de prendre en compte les avis de différents experts simultanément est de réaliser un vote à la majorité. Avec ce modèle, on suppose que tous les experts ont le même niveau et l'on ne cherche donc pas à discriminer les plus compétents. Nous nous attendons donc à obtenir un classifieur aux capacités inférieurs à celles du meilleur annotateur.

1.2 Etat de l'art

De nombreux travaux ont été réalisés sur l'estimation du *ground truth* label à partir de réponses de multiples annotateurs. La majorité des travaux précurseurs furent publiés dans les domaines des biostatiques et de l'épidémiologie. En 1979, Dawid et Skiene [2] proposaient une approche pour estimer le taux d'erreur de multiples annotateurs en fonction des labels qu'ils donnaient. Cependant, comme la plupart des articles de cette époque sur ce sujet, celui se focalise uniquement sur l'estimation du *ground truth* label non observé. C'est seulement plus tard, que des chercheurs se sont intéressés au problème spécifique d'apprendre un classifieur grâce aux données multi-labels. En effet, en 1995 Smyth et al. [3] développent une approche similaire à celle de Dawid et Skiene [2] puis entrainent un classifieur sur cette estimation du *ground truth label*.

Plus récemment et avec la popularité croissante de Amazon Mechanical Truck (AMT) et d'autres plateformes de crowdsourcing, les chercheurs ont commencé à reconnaître l'intérêt du problème d'apprentissage sur des labels de plusieurs anotateurs non-experts. Ainsi en 2010, Raykar et al [4] ont proposé une approche probabiliste innovante où les *ground truth label* et le classifieur sont jointement appris. D'autres travaux ont relaché d'autres hypothèses, tel Yan et al. (2010)[5] qui ne supposent plus que la qualité des labels fournit par les annotateurs ne dépendent pas des instances qu'ils annotent. Ces travaux ont ainsi inspiré de nombreuses variations et extensions dans les dernières années.

2 Modèle

On considère ici que nous disposons d'un ensemble de données de tests X associé à l'ensemble des avis donnés par les experts Y. Ces experts ne possèdent pas tous le même degré d'expertise. Cette notion d'expertise sera toutefois spécifiée dans ce rapport car elle varie en fonction des articles. Le ground truth Z est toutefois une inconnue du problème que l'on cherchera à estimer à l'aide des données du problème.

2.1 Notations

Nous noterons N le nombre de données x_1,...,$x_N \in \mathrm{R}_d$. Pour chacune de ces données, nous avons R labels donnés par R annotateurs. Nous noterons y_i^j le label à la donnée i par l'annotateur j. Le vrai label pour la donnée i est noté z_i.

2.2 Données

Plusieurs jeux de données ont été utilisés pour tester les algorithmes implémentés plus-bas. Ceux-ci sont libre et disponible sur la plateforme de données en ligne UCI Machine Learning Repository: Ionosphere (351,34), Cleveland Heart (297,13), Glass (214,9), and Housing (506,13) (avec (nombre de points, nombre de composantes) chacun). Ces données ne possèdent pas de multiples labels et donc une étape préliminaire de simulation de plusieurs annotateurs est nécessaire; celle-ci varie en fonction des articles.

Nous avons rencontré des difficultés pour tester les algorithmes sur des données réelles puisque ces données sont rares et propriétaires.

3 Learning from crowds

Le modèle que nous avons utilisé a été défini dans l'article [4] de nos références.

3.1 Un modèle à deux pièces pour les annotateurs :

Dans ce modèle, on considère que les annotateurs n'ont pas tous le même niveau d'expertise et qu'à chaque expert on peut associer une sensibilité (taux de vrai positif) et une spécificité (taux de vrai négatif) qui lui est propre. Toutefois, l'hypothèse forte ici formulée est que la probabilité d'erreur de l'expert est indépendante de la donnée x qu'on lui demande de labeliser. Le comportement d'un expert peut donc être entièrement décrit par sa sensibilité :

$$\alpha^j = P_r(y^j = 1|z = 1)$$

et à sa spécificité :

$$\beta^j = P_r(y^j = 0|z = 0)$$

Le complexité de ce problème réside dans le fait que pour l'entrainement des classifieurs, on connait uniquement les labels attribués aux données par les experts mais pas leurs sensibilité et spécificité, ni le *ground truth*.

3.2 La résolution d'un problème en présence de variables manquantes.

La méthode choisie est d'utiliser l'algorithme EM, un algorithme itératif qui permet d'évaluer les paramètres du problèmes particulièrement utile lorsqu'on est en présence de variables cachées.
Les probabilités des deux classes sont modélisées par une méthode de regression logistique. On considère la famille de fonctions linéaires $\mathcal{F} = f_w$ où pour tout w, x de $\mathbb{R}^d$, $f_w(x) = w^T x$.
La probabilité de la classe positive est donc modélisée comme une sigmoid logistique soit $\mathbb{P}[y = 1|x, w] = \sigma(w^T x)$ où la fonction sigmoid logistique est $\sigma(z) = 1/(1 + e^{-z})$.
Nous cherchons à calculer l'estimateur du maximum de vraisemblance de paramètre $\theta = \{w, \alpha, \beta\}$.
Sachant que $\mathbb{P}[X, Y|\theta] = \prod_{i=1}^{N} \mathbb{P}[y_i^1, ..., y_i^R|x_i, \theta]$, en supposant les z_i connus, et le fait que les y_i^j sont conditionnellement indépendants sachant α^j, β^j, et y_i, on peut décomposer la vraisemblance comme suit :

$$\mathbb{P}[X, Y|\theta] = \prod_{i=1}^{N} (\quad \mathbb{P}[y_i^1, ..., y_i^R|y_i = 1, \alpha]\mathbb{P}[y_i = 1|x_i, w] \quad + \quad \mathbb{P}[y_i^1, ..., y_i^R|y_i = 0, \beta]\mathbb{P}[y_i = 0|x_i, w] \quad)$$

Nous supposons que les annotateurs prennent leurs décisions indépendamment (les y_i^1,...,y_i^R sont indépendants). Nous obtenons alors :

$$\mathbb{P}[y_i^1, ..., y_i^R|y_i = 1, \alpha] = \prod_{j=1}^{R} \mathbb{P}[y_i^j|y_i = 1, \alpha^j] \quad = \quad \prod_{j=1}^{R} \left[\alpha^j\right]^{y_i^j} \left[1 - \alpha^j\right]^{1-y_i^j}$$

Et,

$$\mathbb{P}[y_i^1, ..., y_i^R | y_i = 0, \beta] = \quad \prod_{j=1}^{R} \left[\beta^j\right]^{1-y_i^j} \left[1-\beta^j\right]^{y_i^j}$$

Nous pouvons donc écrire la vraisemblance sous la forme : $\mathbb{P}[X, Y|\theta] = \prod_{i=1}^{N} (a_i p_i + b_i(1-p_i))$ où :

$$p_i = \sigma(w^T x_i)$$

$$a_i = \quad \prod_{j=1}^{R} \left[\alpha^j\right]^{y_i^j} \left[1-\alpha^j\right]^{1-y_i^j}$$

$$b_i = \quad \prod_{j=1}^{R} \left[\beta^j\right]^{1-y_i^j} \left[1-\beta^j\right]^{y_i^j}$$

L'estimateur du maximum de vraisemblance se trouve avec la maximisation de la log-vraisemblance :

$$\widehat{\theta} = \left\{\widehat{\alpha}, \widehat{\beta}, \widehat{w}\right\} = argmax_\theta \left\{ln\mathbb{P}[\{X, Y\}|\theta]\right\}$$

Nous allons utiliser l'algorithme EM pour ce problème de maximisation.
Si l'on connait les *ground truth*, on peut écrire la vraisemblance sous la forme :

$$ln\mathbb{P}[X, Y, Z|\theta] = \sum_{i=1}^{N} z_i ln(p_i a_i) + (1-z_i) ln(1-p_i) b_i$$

Chaque itération de l'algorithme consiste en deux étapes :
Etape E : on calcule l'espérance conditionnelle de la vraisemblance à partir de l'estimation courante de θ :

$$\mathbb{E}\left\{lnPr[X, Y, Z|\theta]\right\} = \sum_{i=1}^{N} \mu_i ln(p_i a_i) + (1-\mu_i) ln(1-p_i) b_i$$

avec $\mu_i = \mathbb{P}[z_i | y_i^1, ..., y_i^R, x_i, \theta]$

D'après le théorème de Bayes, on peut écrire :

$$\mu_i = \frac{a_i p_i}{a_i p_i + b_i(1-p_i)}$$

Etape M : A partir de l'estimation courante de μ_i, les paramètres de θ sont obtenus par maximisation de l'espérance conditionnelle. On trouve les valeurs actualisées de α et β quand on résoud gradient de l'espérance conditionnelle égal à 0.

$$\alpha_j = \frac{\sum_{i=1}^{N} \mu_i y_i^j}{\sum_{i=1}^{N} \mu_i}$$

$$\beta_j = \frac{\sum_{i=1}^{N}(1-\mu_i)(1-y_i^j)}{\sum_{i=1}^{N}(1-\mu_i)}$$

Nous utilisons la descente de gradient de Newton-Raphson de la forme $w^{t+1} = w^t - \eta H_{-1} g$ où g est le gradient, H la matrice hessienne et η le pas. Les deux étapes E et M sont itérées jusqu'à la convergence. L'algorithme EM garantit la convergence vers un maximum local.

Il est possible d'aller plus loin avec une approche bayesienne. Nous imposons alors des distributions à priori pour chaque expert. La distribution adoptée dans la suite de l'article est la distribution beta. Une telle approche est intéressante si on sait à priori que certains experts sont plus compétents que d'autres.

3.3 Tests et résultats

3.3.1 Génération de données :

Nous avons généré un ensemble de données perturbées à l'aide des données de base ionosphère que nous avions présentées précédemment. Pour se faire, nous avons considéré que nous disposions de 5 experts aux capacités différentes.
Chaque expert peut donc être modéliser par un couple (α, β).

Considérons un cas donné x. Si le *ground truth* associé est 1 on tire aléatoirement un réel entre 0 et 1. Si l'entier est plus petit que α alors l'expert attribue le label 1, sinon 0. Nous procédons de manière analogue dans le cas où le *ground truth* est nul. Nous faisons alors de même pour chaque expert, et chaque exemple de la base d'apprentissage. L'hypothèse d'indépendance dans les avis donnés par les experts est ici matérialisé par l'indépendance des tirages aléatoires.
Les data étant construites, nous avons alors pu exploiter l'algorithme EM détaillé ci dessus. Le critère d'arrêt choisi est un critère portant sur l'évolution des estimées de sensibilité et spécificité.

3.3.2 Résultats :

L'ensemble de l'implémentation a été réalisé en python. Nous utilisons le fichier tools.py (donné en TP), la bibliothèque sklearn et scipy. Dans une première expérimentation, nous avons considéré un ensemble d'experts aux capacités satisfaisantes. Ainsi l'ensemble des sensibilités et spécificités des annotateurs sont

$$\alpha = (0.75, 0.8, 0.85, 0.9, 0.95)$$

$$\beta = (0.5, 0.55, 0, 6, 0.65, 0.7)$$

Nous obtenons une convergence de l'algorithme après 41 exécutions et nous pouvons dans une première estimation des résultats observer les sensibilités et les spécificités trouvées par l'algorithme. Voici ce que nous obtenons :

$$\alpha_{est} = (0.71, 0.76, 0.77, 0.83, 0.91)$$

$$\beta_{est} = (0.56, 0.61, 0.7, 0.74, 0.82)$$

Ces résultats semblent donc satisfaisants puisque les estimées approchent effectivement les grandeurs théoriques. Nous pouvons de plus remarquer que même si les estimées ne correspondent pas précisément aux données réelles, le classement des experts en fonction de leurs capacités semble quant à lui bien respecter.
Toutefois, ces valeurs ne sont que des intermédiaires de calculs et ne permettent pas de juger de la qualité du classifieur.
Afin d'évaluer le classifieur nous utilisons donc les courbes ROC (Receiver Operating Characteristic). Ces courbes donnent le taux de vrais positifs en fonctions du taux des faux positifs. Elles sont obtenues en faisant varier la valeur de seuil permettant d'attributer un label en fonction de la valeur de $\sigma(w^T x)$.

La qualité d'un classifieur est alors représentée par l'aire sous la courbe (AUC). On peut par exemple constater qu'un classifieur attribuant de manière équiprobables les labels 0 et 1 possédera une courbe ROC suivant la première bissectrice.

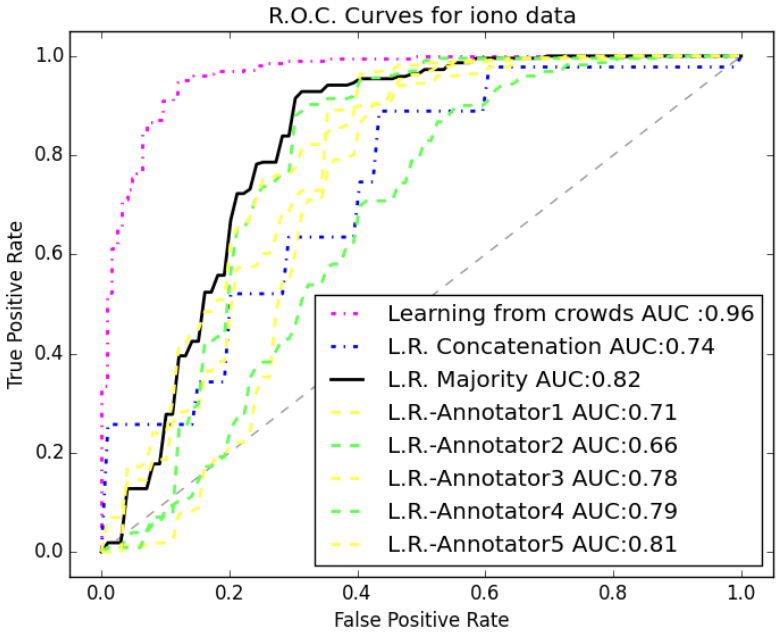


Figure 1: Courbes ROC obtenues par l'approche learning from crowds avec bons annotateurs.

Sur la figure 1, on a fait figurer les courbes ROC associées aux classifieurs construits par une regression logistique des données de chacun des experts, ainsi que les courbes roc obtenues par des méthodes intuitives telles que la concaténation des avis et le vote majoritaire. Les scores de ces classifieurs alternatifs a par ailleurs été mesuré par validation croisée. Les résultats obtenus par la méthode que nous avons ici décrite sont supérieurs à ceux du meilleur des experts ainsi que ceux obtenus par les méthodes plus intuitives que l'on avait envisagé.
Nous pouvons toutefois nous questionner quand aux performances de l'algorithme dans

le cas où les annotateurs sont de moins bonnes qualités : On considère donc un nouvel ensemble d'experts annotant les mêmes données. Les experts sont ici définis par : On obtient ici convergence après 63 itérations.

$$\alpha = (0.4, 0.5, 0.6, 0.7, 0.8)$$

$$\beta = (0.3, 0.4, 0, 5, 0.6, 0.7)$$

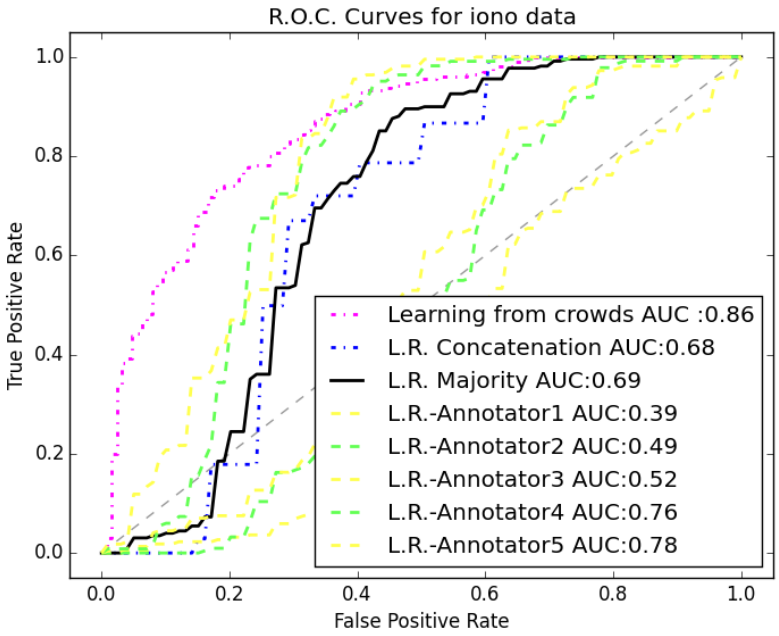


Figure 2: Courbes ROC obtenues par l'approche learning from crowds avec annotateurs relativement peu performants.

Sur la figure 2, a affiché les courbes ROC des différentes méthodes en utilisant les données fournies par des annotateurs de moins bonne qualités. On arrive en effet à des résultats supérieurs à ceux du meilleur expert (AUC de 0.86 contre une AUC de 0.78). On observe ici la faiblesse des méthodes de concaténations (AUC de 0.68) et votes majoritaires (AUC de 0.69) qui ont tendance à produire des résultats peu satisfaisants.

Dans les deux cas les résultats obtenus sont donc très satisfaisants puisqu'ils permettent d'obtenir un score meilleur que celui du meilleur annotateur.
Il faut toutefois relativiser ces résultats car n'ayant pas pu trouver de jeux de données réel nous avons simuler ces dernières en les calquant sur le modèle.

4 Modeling annotator expertise

Dans cet article[5] publié en 2010 par Yan Yan, Rómer Rosales, Glenn Fung, Mark W. Schmidt, Gerardo H. Valadez, LucaBogoni, Linda Moy, et Jennifer G. Dy, les auteurs

relâchent l'hypothèse selont laquelle la qualité des labels fournit par les annotateurs ne dépendent pas des instances qu'ils annotent.

4.1 Modèle

Dans leur modèle, le label fournit par l'annotateur t dépend du *ground thruth* label(inconnu z mais aussi de la données d'entrée x. Les auteurs supposent donc que les annotateurs ont des performances inégales et que celles-ci dépendant de la donnée qu'ils annotent. Ceux-ci supposent en revanche que les annotateurs $t = 1, ..., T$ sont indépendant conditionnellement à la donnée qu'ils annotent et au *ground truth* label. Ainsi

$$p(Y, Z|X) = \prod_i p(z_i|x_i) \prod_t p(y_i^{(t)}|x_i, z_i)$$

Il est supposé que chaque annotateur fournit une version bruitée du *ground truth* label z,

$$p(y_i^{(t)}|x_i, z_i) = (1 - \eta_t(x_i))^{\left|y_i^{(t)} - z_i\right|} \eta_t(x_i)^{1 - \left|y_i^{(t)} - z_i\right|}$$

avec η_t la fonction logistique:

$$\eta_t : x \to (1 + exp\left(-w^\intercal x - \gamma_t\right))^{-1}$$

Dans ce modèle de Bernouilli, le paramètre $\eta_t(x_i)$ est la probabilité que l'annotateur t soit correct, c'est à dire que $y_i^{(t)} = z_i$. Les auteurs proposent une modélisation alternative; ils considèrent un modèle gaussien où chaque annotateur fournit une version déformée du *ground truth* label z:

$$p(y_i^{(t)}|x_i, z_i) = \mathcal{N}\left((y_i^{(t)}; z_i, \sigma^{(t)}(x_i)\right)$$

avec σ_t la fonction logistique:

$$\sigma_t : x \to (1 + exp\left(-w^\intercal x - \gamma_t\right))^{-1}$$

Par simplicité les auteurs choisissent une un modèle logistique pour $p(z_i|x_i)$:

$$p(z_i = 1|x_i) = (1 + exp\left(-\alpha^\intercal x - \beta\right))^{-1}$$

Les paramètre $\theta = \{\alpha, \beta, \{w_t\}, \{\gamma_t\}\}$ sont alors estimés en maximisant la log vraisemblance, en incluant la variable *ground truth* z:

$$\theta_{ML} = \underset{\theta}{argmax} \sum_t \sum_i \log \sum_{z_i} p(y_i^{(t)}, z_i|x_i; \theta)$$

4.2 Implémentation

Puisque le modèle possède une variable latente z, une approche classique est d'employer un algorithme EM (Espérance-Maximisation) pour calculer le maximum de vraisemblance. Ainsi, les paramètres sont approchés itérativement en estimant la distribution a posteriori des *ground truth* labels puis en utilisant cette estimation pour calculer l'argument maximisant la vraisemblance. Ainsi, nous alternons l'étape d'espérance:

Etape E: Calculer

$$\tilde{p}(z_i) = p(z_i|x_i, y_i)$$

$$\tilde{p}(z_i) \quad \propto \quad p(z_i, y_i|x_i) = \prod_t p(y_i^{(t)}|x_i, z_i)p(z_i|x_i)$$

Etape M: Maximiser

$$\sum_t \sum_i E_{\tilde{p}(z_i)} \left[\log p\left(y_i^{(t)}, z_i|x_i\right)\right]$$

$$= \sum_{i,t} E_{\tilde{p}(z_i)} \left[\log p\left(y_i^{(t)}|x_i, z_i\right) + \log p\left(z_i|x_i\right)\right]$$

4.3 Simulation des données

Face à la difficulté de trouver des données réelles avec plusieurs annotateurs, nous avons choisis de tester l'algorithme sur des données simulées. En effet, les données Ionosphere de UCI sont des données avec un label biclasse, il a donc fallut simuler plusieurs annotateurs. Pour ce faire nous avons suivit la même démarche que les auteurs dans l'article. Premièrement, le jeux de données est subdivisé en 5 par la méthode de k-means, puis il est supposé que chacun des 5 annotateurs est expert sur l'un des 5 cluster où leur réponse coïncide avec le *ground truth* label. Pour les autres clusters, il est supposé que l'annotateur t a 35% de chance de se tromper.

4.4 Résultats

Pour évaluer notre classifeur, nous entraînons d'autres classifieurs de regression logistique; un utilisant le label majoritaire comme label pour l'apprentissage, un autre qui concatène les labels des annotateurs et répète les données d'apprentissage, et 5 autres avec chacun les labels donnés par un annotateur. Ainsi, sont présentées sur la figure 3 les courbes ROC des différents classifieurs, et il apparait que l'approche développée dans cet article est concluante puisque son AUC est la plus grande. Cependant, ce résultat est à relativiser puisque il n'est pas tout le temps aussi bon, car la simulation des données est grandement aléatoire et que le résultat final en dépend.

Sur les figures 4 5 nous pouvons observer la contribution moyenne de chaque annotateur pour différents clusters. Les résultats sont satisfaisants puisque le modèle a bien appris quel annotateur était performant sur quel partie des données.

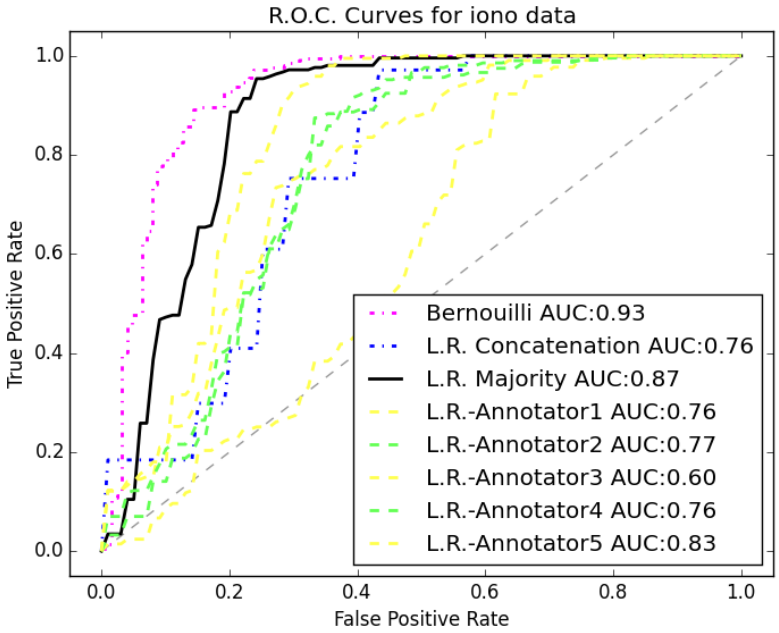


Figure 3: Courbes ROC obtenues pour le jeux de données Ionosphere

Références

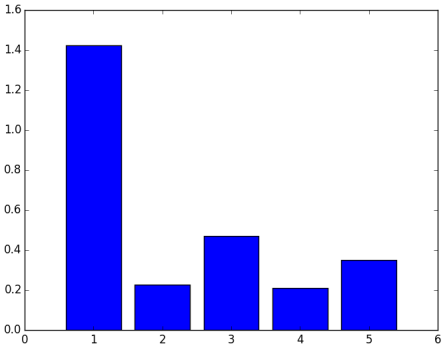
[1] Jeff Howe. *Crowdsourcing: Why the Power of the Crowd Is Driving the Future of Business*. Crown Publishing Group, New York, NY, USA, 1 edition, 2008.

[2] P. Dawid, A. M. Skene, A. P. Dawidt, and A. M. Skene. Maximum likelihood estimation of observer error-rates using the em algorithm. *Applied Statistics*, pages 20–28, 1979.

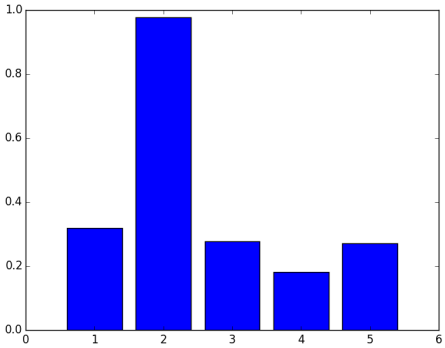
[3] Padhraic Smyth, Usama M. Fayyad, Michael C. Burl, Pietro Perona, and Pierre Baldi. Inferring ground truth from subjective labelling of venus images. In G. Tesauro, D.S. Touretzky, and T.K. Leen, editors, *Advances in Neural Information Processing Systems 7*, pages 1085–1092. MIT Press, 1995.

[4] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, and Linda Moy. Learning from crowds. *J. Mach. Learn. Res.*, 11:1297–1322, August 2010.

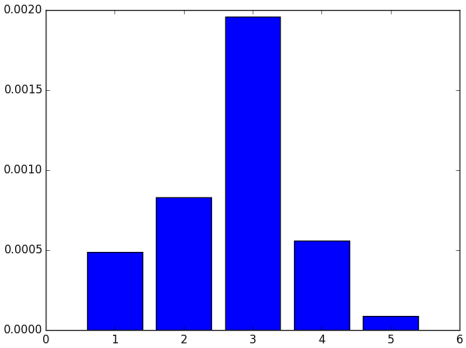
[5] Yan Yan, Rómer Rosales, Glenn Fung, Mark W. Schmidt, Gerardo H. Valadez, Luca Bogoni, Linda Moy, and Jennifer G. Dy. Modeling annotator expertise: Learning when everybody knows a bit of something. *Journal of Machine Learning Research - Workshop and Conference Proceedings*, 9:932–939, 2010.



(a) Cluster 1

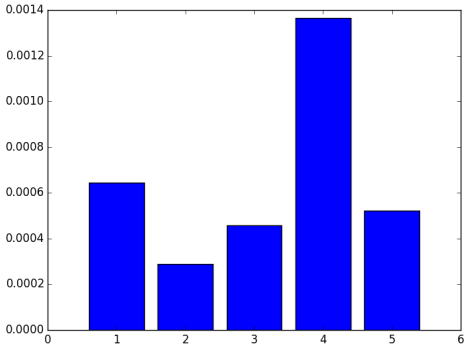


(b) Cluster 2

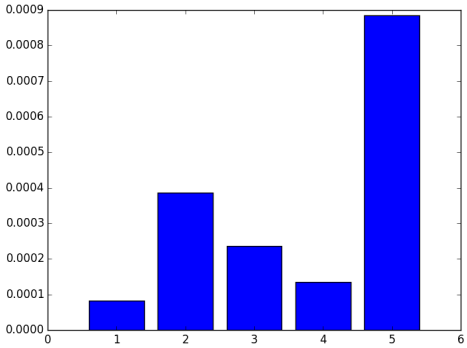


(c) Cluster 3

Figure 4: Contribution moyenne des 5 annotateurs pour un cluster donnée



(a) Cluster 4



(b) Cluster 5

Figure 5: Contribution moyenne des 5 annotateurs pour un cluster donnée

——— *Fin du compte-rendu* ———