Rapport du projet de Machine Learning

Learning from the Crowd

Corvisier Jean-Christophe
Deloro Yonatan
Kheldouni Mohammed Amine

24 juin 2017

Motivations

Ce projet s'inscrit dans le champ de l'apprentissage supervisé, non pas par un professeur qui dicte les bons labels, mais par différents annotateurs plus ou moins fiables. En effet, pour certains types de problèmes, la vérité terrain est très difficilement accessible (dans le domaine médical, effectuer la tomographie d'un tissu pour détecter la présence d'une tumeur nécessite en pratique une biopsie qui n'est pas une procédure sans risques) ou alors cette vérité n'est tout simplement pas clairement définie (pour des tacĥes plus subjectives comme des jugements d'opinion, si l'on souhaite par exemple apprendre à un robot à déterminer si un sourire est forcé ou non).

Ainsi on dispose au mieux d'étiquettes données par différents annotateurs qui sont par essence bruitées voire fausses : si les experts en radiologie donnent des diagnostics différents, parfois on ne dispose pas même d'experts mais d'une foule de personnes donnant leur ressenti sur une tâche subjective ou apportant leur propre savoir sur un sujet donné, comme sur Wikipedia par exemple. La toujours plus grande facilité de partager les données et le mouvement OpenSource par exemple créent par nature de telles situations de *crowdsourcing* où l'on dispose de très nombreux labels ou étiquetages bruitées et non pas d'un seul expert qui joue le rôle de professeur ou de vérité terrain comme dans le cadre classique de l'apprentissage supervisé.

Objectifs

L'idée du projet consiste donc à combiner les savoirs des différents annotateurs pour labelliser mieux que les avis séparés des experts ou que le jugement moyen ou majoritaire d'une grande quantité d'annotateurs amateurs. Ce en essayant d'apprendre, conjointement à la manière de prédire de ces différents annotateurs, leur niveaux de connaissance ou de fiabilité selon la donnée, et leurs éventuelles corrélations.

Plus particulièrement, nous nous focaliserons sur des modèles de *Crowdlearning* permettant de prédire des labels binaires +1 ou 0 à partir des variables descriptives des données (*features*), grâce à un jeu de données d'entraînement constitué de leurs *features* et de leurs labels attribués par différents annotateurs.

Méthodologie suivie

Ainsi, on implémentera dans un premier temps un premier modèle de *Crowdlearning*, le modèle de classification binaire appris sur annotateurs multiples de Raykar, Yu et al, qui fait l'hypothèse d'un certain niveau de fiabilité de chaque annotateur, fiabilité caractérisée par leurs sensibilités (taux de vrais positifs) et leurs spécificités (taux de vrais négatifs) supposés uniformes sur l'ensemble des données. Dans un deuxième temps, on se penchera sur un modèle plus fin qui fait l'hypothèse que cette fiabilité dépend de la donnée, autrement dit que les annotateurs ont a priori une connaissance plus importante dans certaines régions de l'espace des données que dans le reste de l'espace.
Une telle modélisation supposant donc l'existence de domaines d'expertises des annotateurs se fondera sur l'article de Yan, Rosales et al, et fera l'hypothèse, pour diminuer le nombre de paramètres, d'une égalité entre spécificité et sensibilité d'un annotateur dans une même région des données. Enfin, dans un dernier temps, on esquissera un modèle où les annotateurs ne sont plus indépendants conditionnellement à la donnée, chacun étant supposé plus ou moins sensible à une consigne extérieure dictant un label à attribuer. Nous testerons d'abord ces modèles sur des données artificielles pour évaluer ensuite leur pertinence sur l'*Adult dataset* de UCI Machine Learning, décrivant 30.000 d'individus par différentes caractéristiques socio-économiques, collectant leurs salaires faisant office de labels, et confrontant ceux-ci aux prédictions de

11 économistes ayant accès aux différentes *features*.

L'intégralité du code - génération de données artificielles et extraction des données réelles, implémentation des différents modèles de *Crowdlearning*, fonctions de visualisation des résultats, et expériences -, a été implémenté en PYTHON, nous en joignons les briques essentielles au projet.

Plan du rapport

1 Notations du problème

Dans ce problème de *Crowdlearning*, nous supposerons donc avoir à disposition un ensemble d'apprentissage composé d'un certain nombre N de données $X_i, i \in [1, N]$ décrites par D dimensions : $x_i \in \mathbb{R}^D$. Ainsi nous noterons $X = (x_1...x_n)^T$ la matrice des *features* de taille (N, D).

Pour chacune de ces données, on disposera également des labels donnés par T annotateurs. On notera ainsi $y_i^t \in \{0, 1\}$ le label délivré par l'annotateur $t \in [1, T]$, et Y la matrice des annotations de taille (N, T), définie par : $Y = \begin{pmatrix} y_1^1 & ... & y_1^T \\ y_2^1 & ... & y_2^T \\ \vdots & \ddots & \vdots \\ y_N^1 & ... & y_N^T \end{pmatrix}$.

Enfin on notera $z_i \in \{0, 1\}$ le vrai label correspondant à la donné x_i, et Z le vecteur colonne les listant. Évidemment, nous supposerons avoir aucune connaissance de Z pour l'apprentissage, que nous introduisons pour des soucis de notation dans la construction du modèle.

Ainsi, connaissant l'ensemble d'apprentissage renommé X_{train}, Y_{train}, il s'agira de prédire la relation liant le vrai label z aux *features* x d'une donnée. D'autres prédictions sont possibles : z à partir de x et des diverses annotations y, ou encore à partir de y uniquement grâce à de l'échantillonnage sur le jeu d'entraînement. Aussi nous nous limiterons dans le cadre du projet à prédire les vrais labels Z à partir des données brutes X. Nous confronterons nos résultats aux vrais labels Z, la vérité terrain étant connue pour les données que nous avons utilisées.

2 L'hypothèse d'un niveau de fiabilité a priori de chaque annotateur : le modèle de Raykar, Yu et al [1]

2.1 Description du modèle

Le premier modèle de *Crowdlearning* que nous avons considéré et implémenté est celui développé par Raykar, Yu et al dans [1].

Pour des raisons de simplicité, le modèle choisit de classifier Z linéairement en fonction de X, par régression logistique, même si des fonctions d'un autre type seraient également envisageables. Ainsi, on cherchera un classifieur de la forme $f_w(x) = w^T x + \gamma = W^T \widetilde{x}$, où $\widetilde{x} = (1, x)$ et $W \in \mathbb{R}^{d+1}$. Et la probabilité du label 1 sachant la donnée x_i pourra alors être estimée en composant f_W par la fonction sigmoïde ainsi que :

$$\mathbb{P}(Z_i = 1 | x_i, W) = \frac{1}{1 + e^{-W^T \widetilde{x_i}}} = \sigma(W^T \widetilde{x_i})$$

On attribuera alors à x_i le label +1 si $\mathbb{P}(Z_i = 1 | x_i, W)$ est supérieur à un certain seuil fixé γ, et le label 0 dans le cas contraire.

Dans ce modèle les annotateurs sont supposés indépendants. Aussi, afin de pouvoir donner plus de poids à un annotateur qu'à un autre et obtenir ainsi de meilleures performances qu'un simple jugement majoritaire, le modèle fait l'hypothèse importante que chaque

annotateur t a un certain niveau de fiabilité. Pour ce faire, il caractérise ainsi chaque annotateur par sa probabilité α_t d'énoncer à raison le label 1 (sensibilité, taux de vrais positifs), et par sa probabilité β_t d'énoncer à raison le label 0 (spécificité, taux de vrais négatifs). Ainsi on a (avec la formule des probabilités totales) :

$$\mathbb{P}(Y_i^t = 1) = r_i\alpha_t + (1 - r_i)(1 - \beta_t)$$

avec $r_i = \mathbb{P}(Z_i = 1)$

α_t et β_t sont donc des variables du modèle, en supplément des poids W du classifieur. L'algorithme proposé recherche ainsi ces paramètres α_t, β_t en même temps que les poids du classifieur de manière à maximiser la vraisemblance du modèle : "heuristiquement", cela lui permet d'augmenter le poids d'un annotateur dans la classification s'il le juge meilleur que les autres.

Plus précisément, la vraisemblance du modèle s'écrit, en posant $\theta = (\alpha, \beta, W)$, et en supposant les exemples indépendants entre eux :

$$\theta \to L(X, Y|\theta) = \prod_{i=1}^{N} \mathbb{P}(\bigcap_{1\leq t\leq T} \{Y_i^t = y_i^t\}|x_i, \theta)$$

En utilisant l'indépendance des annotations Y_i^t conditionnellement à α, β, Z_i de toute autre variable dont la donnée X_i et les poids w du classifieur, et supposant l'indépendance mutuelle des Y_i^t conditionnellement à Z_i, la vraisemblance se réécrit :

$$\theta \to L(X, Y|\theta) = \prod_{i=1}^{N} a_i p_i + b_i(1 - p_i)$$

où :

$$\begin{cases} p_i &= \mathbb{P}(Z_i = 1|x_i, W) = \sigma(W^T \widetilde{x_i}) \\ a_i &= \mathbb{P}(\bigcap_{1\leq t\leq T} \{Y_i^t = y_i^t\}|Z_i = 1, \alpha) \\ &= \prod_{t=1}^{T} \alpha_t^{y_i^t}(1 - \alpha_t)^{(1-y_i^t)} \\ b_i &= \mathbb{P}(\bigcap_{1\leq t\leq T} \{Y_i^t = y_i^t\}|Z_i = 0, \beta) \\ &= \prod_{t=1}^{T} \beta_t^{(1-y_i^t)}(1 - \beta_t)^{y_i^t} \end{cases}$$

Ainsi il s'agit de trouver les paramètres $\theta^\star = (\alpha^\star, \beta^\star, W^\star)$ qui vont maximiser la vraisemblance du modèle.

2.2 Algorithme

On effectue pour ce faire un algorithme d'EM (*Expected Maximization*) où les vrais labels Z jouent le rôle de données cachées.

Si l'on connaissait les vrais labels Z, la vraisemblance du modèle s'écrirait ainsi :

$$\theta \to L(X, Y, Z|\theta) = \prod_{i=1}^{N} (a_i p_i)^{z_i} (b_i(1 - p_i))^{1-z_i}$$

Ou encore la log-vraisemblance :

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

L'algorithme EM est un algorithme itératif qui à chaque itération maximise une borne inférieure de la vraie log-vraisemblance, donnant alors une convergence vers un maximum local. La borne inférieure utilisée est l'espérance de la log-vraisemblance conditionnellement à $\widetilde{p}(Z) = P(Z|X, Y, \underline{\theta})$ probabilités des vrais labels étant donné les observations et le paramètre $\underline{\theta}$ courant. A chaque itération de l'EM, nous procédons en deux étapes :

Lors de l'"E-step", on calcule la fonction log-vraisemblance conditionnelle à $P(Z|X, Y, \theta)$ et espérée sachant les observations X, Y et le paramètre $\underline{\theta}$ courant :

$$\theta \to \mathbb{E}_{\widetilde{p}(Z)} ln(L(X, Y, Z|\theta)) = \sum_{i=1}^{N} \widetilde{p_i} \ln a_i p_i + (1 - \widetilde{p_i}) \ln b_i(1 - pi)$$

où :

$$\widetilde{p_i} = \mathbb{P}(Z_i = 1|x_i, \bigcap_{t=1} \{Y_i^t = y_i^t\}, \underline{\theta})$$

est proportionnelle via la règle de Bayes à

$$\mathbb{P}(\bigcap_{t=1} \{Y_i^t = y_i^t\}|Z_i = 1, \underline{\theta})\mathbb{P}(Z_i = 1|x_i, \underline{\theta})$$

soit :

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

Cela nous permet donc lors de l'E-Step de mettre à jour les valeurs des $\widetilde{p_i}$ étant donné le paramètre $\underline{\theta}$ courant.

Lors de la "M-step", on cherche à maximiser l'espérance conditionnelle décrite précédemment, ce qui revient alors à chercher

$$\theta^\star = argmax[\mathbb{E}_{\widetilde{p}(Z)}(ln(L(X, Y, Z|\theta))]$$

$\alpha^\star$ et $\beta^\star$ s'obtiennent par formes closes, puis w est optimisé grâce à une descente de gradient sur l'opposé de la fonction, avec l'algorithme de Newton-Raphson (voir l'annexe pour les expressions de $\alpha^\star$, $\beta^\star$ et les formules des gradients et hessiennes en W).

On alterne les deux étapes jusqu'à convergence vers un maximum local.

Enfin, notons que, pour l'implémentation, il faut initialiser les $\widetilde{p_i} = \mathbb{P}(Z_i|x_i, y_i, \theta)$ ce que l'on fait par jugement majoritaire : $\widetilde{p_i^0} = \mathbb{P}(Z_i|y_i) = \frac{1}{T}\sum_{t=1}^{T} y_i^t$. Aussi il faudra initialiser w dans la descente de gradient : à la première itération on l'initialisera donc aléatoirement, pour les suivantes on prendra le w optimisé à la précédente itération.

Conséquence intéressante du modèle En prenant le *logit* des probabilités des labels estimées "a posteriori" $\widetilde{p_i} = \mathbb{P}(Z_i = 1|x_i, y_i, \theta)$, on obtient :

$$\begin{aligned}\ln \frac{\mathbb{P}(Z_i=1|x_i,y_i,\theta)}{\mathbb{P}(Z_i=0|x_i,y_i,\theta)} = \quad & W^T \widetilde{x_i} + Cte + \\ & \sum_{t=1}^{T} Y_i^t (logit[\alpha_t] + logit[\beta_t])\end{aligned}$$

Où $Cte = \sum_{t=1}^{T} \ln \frac{1-\alpha_t}{\beta_t}$ ne dépend pas de la donnée.

Ainsi une conséquence intéressante du modèle est que la frontière de classification se caractérise donc par une combinaison linéaire des *features* pondérée par les poids (W) appris ainsi qu'une combinaison linéaire pondérée des labels des annotateurs, où le poids de chaque annotateur est égale à la somme des logits de sa sensibilité et de sa spécificité.

3 Des annotateurs spécialisés dans certains domaines : le modèle de Yan, Rosales et al. [2]

Jusque là nous avons supposé que les annotateurs avaient la même probabilité de se tromper quelle que soit la donnée. Or pour des données suffisamment riches, à savoir un espace des données assez grand pour lequel un apprentissage commence à devenir intéressant, il est assez probable que le niveau de connaissance a priori des annotateurs dépende du lieu où se situe la donnée. Et on peut supposer que, si tel est le cas, il y a bien plus à tirer de l'approche *Crowdlearning* (on peut espérer d'encore meilleures prédictions par rapport au jugement majoritaire) en combinant les savoirs de plusieurs annotateurs différemment "cultivés" que si un annotateur est strictement meilleur ou moins bon qu'un autre quelque soit la région de la donnée.

En pratique, on peut se convaincre, pour de nombreuses applications, que les annotateurs puissent avoir une expertise spécialisée dans certaines régions des données. Si pour analyser une radio certains médecins seront spécialistes de telle ou telle type de pathologie, les modifications d'un certain article de Wikipedia par un contributeur sont peut-être moins fiables sans que tous les articles qu'il ait écrits ne doivent être soumis à correction.

3.1 Description du modèle

C'est dans cette volonté de modéliser plus finement la réalité et de tirer parti de manière plus significative du principe de *crowdleraning* que s'inscrit le modèle développé par Yan, Rosales et al. [2] : "*Modeling anotator expertise*". Aussi comme la fiabilité du label de l'annotateur dépend maintenant de la donnée qu'il annote, on ne différencie plus spécificité et sensibilité pour ne pas augmenter le nombre de paramètres déjà important à déterminer dans le modèle. Désormais on appellera η_t la probabilité de l'annotateur t à donner le bon label (il donne le mauvais label avec une probabilité $1 - \eta_t$) qui sera donc fonction de la donnée x.

Et on supposera simplement que cette probabilité, ou encore la connaissance de l'annotateur t, $x \to \eta_t(x)$, a la forme d'une fonction logistique en fonction de la donnée :

$$\eta_t(x) = \frac{1}{1 + \exp(-\alpha_t^T x - \beta_t)}$$

avec $\alpha_t \in \mathbb{R}^d$ et $\beta_t \in \mathbb{R}$ (on a gardé les notations α_t et β_t pour qualifier des paramètres caractérisant la connaissance de l'annotateur t, mais ils ne jouent plus ici les rôles de sensibilité et de spécificité).

Ainsi on pourra écrire :

$$\mathbb{P}(y_i^t|z_i, x_i, \alpha_t, \beta_t) = \eta_t(x_i)^{|y_i^t - z_i|}(1 - \eta_t(x_i))^{1-|y_i^t - z_i|}$$

On cherche toujours un classifieur de la forme $f_w(X) = w^T X + \gamma$, avec $\mathbb{P}(Z = 1|x, w, \gamma)$ s'exprimant toujours par l'action de la sigmoïde sur $f_w(X)$.

Pour déterminer celui-ci, on maximise donc toujours la log-vraisemblance du modèle par rapport à $\theta = \{\alpha_t, \beta_t, w, \gamma\}$, laquelle s'écrit avec l'hypothèse d'indépendance mutuelle des Y_i^t conditionnellement à θ :

$$\begin{aligned}\theta \to L(X, Y|\theta) &= \prod_{i=1}^{N} \mathbb{P}(\bigcap_{1 \le t \le T} \{Y_i^t = y_i^t\}|x_i, \theta) \\ &= \prod_{i=1}^{N} \prod_{t=1}^{T} \mathbb{P}(y_i^t|x_i, \theta)\end{aligned}$$

3.2 Algorithme

On effectue toujours pour ce faire un algorithme d'EM.

En introduisant les variables cachées Z_i, la vraisemblance du modèle s'écrit ainsi :

$$\theta \to L(X, Y, Z|\theta) = \prod_{i=1}^{N} \prod_{t=1}^{T} \sum_{z_i \in \{0,1\}} \mathbb{P}(y_i^t, z_i|x_i, \theta)$$

Ainsi on calcule lors de l'"E-step" la fonction log-vraisemblance conditionnelle à $\widetilde{p}(Z) = P(Z|X, Y, , \underline{\theta})$ et espérée sachant les observations X, Y et le paramètre $\underline{\theta}$ courant :

$$\begin{aligned}&\theta \to \mathbb{E}_{\widetilde{p}(Z)} ln(L(X, Y, Z|\theta)) \\ &= \sum_{i=1}^{N} \sum_{t=1}^{T} \sum_{z_i=0}^{1} \widetilde{p}(z_i) \ln \mathbb{P}(y_i^t, z_i|x_i, \alpha_t, \beta_t)\end{aligned}$$

où :

$$\widetilde{p}(z_i) = \mathbb{P}(z_i|x_i, \bigcap_{1\leq t\leq T}\{Y_i^t = y_i^t\}, \underline{\theta})$$

est proportionnelle via la règle de Bayes à

$$\begin{aligned}&\mathbb{P}(\bigcap_{1\leq t\leq T}\{Y_i^t = y_i^t\}|z_i, x_i, \underline{\theta})\mathbb{P}(z_i|x_i, \underline{\theta})\\ &= \prod_{t=1}^{T}\mathbb{P}(y_i^t|z_i, x_i, \alpha_t, \beta_t)\mathbb{P}(z_i|x_i, w, \gamma)\end{aligned}$$

Lors de la M-step, on calcule alors :

$$\theta^\star = argmax[\mathbb{E}_{\widetilde{p}(Z)}(ln(L(X, Y, Z|\theta))]$$

Il n'y a pas de forme close pour ces paramètres, ainsi on utilise un algorithme de gradient à pas variable pour déterminer $((d + 1)(T + 1)$ variables réelles à optimiser dont les expressions des dérivées partielles sont explicitées en annexe).

L'algorithme impose d'initialiser θ. L'algorithme EM ne garantit une convergence que vers un maximum local, aussi comme nous l'avons vu, modifier l'initialisation change la convergence de l'algorithme. Nous avons alors choisi de prendre les valeurs de ses paramètres aléatoirement entre 0 et 1.

Conséquences du modèle - Une conséquence similaire à celle obtenue dans le premier modèle s'obtient en prenant le *logit* des $\widetilde{p_i} = \mathbb{P}(Z_i = 1|x_i, y_i, \theta)$:

$$\ln\frac{\mathbb{P}(Z_i = 1|x_i, y_i, \theta)}{\mathbb{P}(Z_i = 0|x_i, y_i, \theta)} = w^T x_i + \gamma + \sum_{t=1}^{T}(-1)^{1-y_i^t}(\alpha_t^T x_i + \beta_t)$$

La contribution d'un annotateur t à la classification est donc donné par le modèle d'annotation spécifique à cet annotateur (α_t, β_t), pondérée positivement ou négativement selon le label attribué.

4 Expériences et résultats des deux modèles

4.1 Données étudiées

Dans un premier temps, on testera les différents modèles sur deux jeux de données artificielles pour se convaincre de leur validité :

— un premier jeu de données artificielles 2D (dénommé ci après jeu 1) : deux gaussiennes centrées en deux points éloignés sur la droite $y = x$, labellisées respectivement $+1$ et 0, que l'on bruite avec un niveau ϵ pour tester les cas séparable et non séparable.

— un second jeu de données 2D (dénommé ci après jeu 2), quatre gaussiennes centrées sur les quatre demi-axes du plan $((1, 0), (-1, 0), (0, 1), (0, -1))$, les gaussiennes au dessus de la droite $y = -x$ étant labellisées $+1$, celles en dessous étant labellisées -1 (voir la figure 7). Ce deuxième jeu permettra de tester le gain que permet potentiellement d'atteindre le deuxième modèle où les annotateurs sont supposés spécialisés dans certaines régions des données.

Pour générer les différents labels, on considérera :

— dans un premier temps des annotateurs de Bernoulli ayant même probabilité de succès quelque soit la donnée. On considérera d'abord une population d'annotateurs tous moyennement fiables, puis on augmentera progressivement leur fiabilité notamment pour comparer les performances du premier modèle de *Crowdlearning* au jugement majoritaire.

— dans un second temps, des annotateurs dont la qualité dépend du lieu de la donnée. En particulier, pour confronter les deux modèles de *Crowdlearning* (non spécialisé et spécialisé), on considérera donc un annotateur spécialisé dans la région $|y| \geq |x|$ et un autre dans la région $|y| \leq |x|$ (domaine jaune), pour le deuxième jeu de données artificielles (voir la figure 7).

Après test des modèles sur ces différents jeux de données artificielles, on testera leur pertinence sur l'Adult dataset de UCI Machine Learning, qui décrit 30.000 d'individus par différentes caractéristiques socio-économiques, qui formeront notre espace de features X, par leurs salaires qui donneront les labels Z à prédire ($Z = +1$ si le salaire est supérieur à 50K$, 0 sinon), et qui confrontent ces salaires aux prédictions de 11 économistes ayant accès aux différentes caractéristiques, qui correspondront donc à la matrice Y.

4.2 Méthodologie

Aussi bien pour les données artificielles que pour les données réelles, nous chercherons à prédire, étant données les dimensions descriptives d'un jeu de données X_{test}, leurs labels Z_{test}, ce grâce à un ensemble d'apprentissage X_{train}, Y_{train}.

Pour les données artificielles, nous générerons deux ensembles de tailles équivalentes dont nous préciserons la taille dans chaque expérience. Pour les données réelles, nous prendrons un découpage train/test de 0.8/0.2 dans nos expériences (après permutation). Bien entendu, nous aurons effectué au préalable un traitement nécessaire de centrage-réduction des données.

Aussi bien pour les données artificielles que pour les données réelles, nous nous attacherons, avant toute interprétation, à vérifier la convergence de l'algorithme

EM. Nous paierons également attention aux risques de sur-apprentissage dans le problème avec données réelles même si le nombre de dimensions reste relativement modeste.

Enfin, nous comparerons les prédictions des labels des différents modèles de *Crowdlearning* à partir des *features* X aux prédictions de deux autres classifieurs :

— un simple jugement majoritaire, qui choisit le label voté par la majorité des annotateurs, en les considérant donc tous comme identiquement fiables. Il s'agit donc d'une prédiction sans apprentissage en amont, et uniquement à partir des différentes annotations Y :

$$\mathbb{P}(Z_i = 1|Y) = \frac{1}{T}\sum_{t=1}^{T} Y_i^t$$

— un classifieur linéaire par régression logistique, qui prédit le label à partir des *features* X et qui est appris en connaissance de la vérité terrain Z_{train} des données d'apprentissage X_{train}.

Plus précisément, le modèle considère que la probabilité $\mathbb{P}(Z_i = 1|X_i, W) = \frac{1}{1+e^{W^T\widetilde{X_i}}} = \sigma(W^T\widetilde{X_i})$ $\widetilde{X_i} = (1, X_i)$ et où $W \in \mathbb{R}^{d+1}$ et, une fois l'apprentissage terminé, attribue le label $+1$ ou 0 selon la position de $\mathbb{P}(Z_i = 1|X_i, W)$ vis-à-vis d'un seuil γ fixé.

Pour l'apprentissage, le modèle choisit le W qui minimise, sur le jeu d'entraînement rebaptisé (X, Z) la fonction de coût empirique $l(W, X, Z)$ qui correspond à l'opposé de la log-vraisemblance (que l'on veut en effet maximale), soit :

$$\begin{aligned} -l(W, X, Z) &= -\sum_{i=1}^{N} \log \frac{1}{1 + \exp -Z^i \widetilde{X}^i W} \\ &= \sum_{i=1}^{N} \log\left(1 + \exp -Z^i \widetilde{X}^i W\right) \end{aligned}$$

On optimisera le choix de W grâce à un algorithme classique de descente de gradient (avec un pas adéquat). Le gradient de la fonction de coût est donnée en annexe.

Ainsi grâce à ces deux comparaisons, on pourra quantifier l'apport des modèles de *Crowdlearning*, c'est-à-dire des hypothèses sur la fiabilité a priori des annotateurs, par rapport aux prédictions du jugement majoritaire, et dans la limite de celles données par un classifieur idéal apprenant sur la vérité terrain, rôle joué par le classifieur par régression logistique appris avec les Z.

Pour comparer les performances entre les modèles de *Crowdlearning* au jugement majoritaire et au classifieur idéal appris avec les vrais labels, nous tracerons des courbes ROC (taux de vrais positifs ou sensibilité, en fonction du taux de faux positifs ou "1 moins la spécificité"), en attribuant les labels, une fois l'apprentissage de chaque modèle terminé, selon différents seuils γ étalés dans l'intervalle $[0, 1]$:

$$\begin{cases} 1 & \text{si} \quad \mathbb{P}(Z_i = 1|X) > \gamma \\ 0 & \text{si} \quad \mathbb{P}(Z_i = 1|X) \leq \gamma \end{cases}$$

4.3 Résultats du premier modèle (non spécialisé) sur données artificielles

Expérience 1 - Dans un premier temps, on teste le premier modèle de *Crowdlearning* sur le jeu 1 de données artificielles, pour 5 annotateurs tous fiables à 0.6.

Dans le cas séparable, on observe sur la figure 1 que le modèle non spécialisé permet de retrouver les bonnes prédictions sur l'ensemble d'apprentissage à l'aide des labels attribués par les annotateurs. On note en effet, qu'au bout de 100 itérations de l'algorithme EM, quand la log-vraisemblance espérée a convergé vers un maximum comme on l'observe sur la figure, les valeurs des paramètres du modèle correspondant aux sensibilités et spécificités des 4 annotateurs sont (en arrondissant) : $\alpha = (0.59, 0.66, 0.66, 0.54, 0.59)$ et $\beta = (0.71, 0.52, 0.54, 0.62, 0.72)$; ce qui montre que l'algorithme a retrouvé des estimations assez proches des fiabilités des annotateurs générés (autour de 0.6). Observant la courbe ROC obtenue pour cette même expérience (figure 2), on observe que l'on obtient, sur l'ensemble d'apprentissage comme sur l'ensemble de test, une classification parfaite avec le premier modèle, comme le classifieur RegLog appris sur les vrais labels, transcendant ainsi les performances du *Majority Voting*. Cette exactitude dans la prédiction est bien entendu liée à la séparabilité des données et au fait que les hypothèses du modèle de *Crowdlearning* correspondent exactement à la manière de générer les annotateurs (annotateurs indépendants de Bernoulli).

Dans le cas non séparable (figure 3), on note également la convergence de l'EM et que, au bout de celle-ci, le modèle de *Crowdlearning* permet de tracer une frontière plutôt pertinente pour tenter de séparer les labels bleus et rouges du Train (en réalité, seuls deux points basculent par rapport à la frontière tracée par le classifieur linéaire par régression logistique appris sur vrais labels Z). On observe ainsi (figure 4) que la courbe ROC du modèle de *Crowdlearning* "talonne" la courbe du classifieur linéaire appris sur la vérité terrain. Ces bonnes performances sont obtenues avec des estimations des fiabilités des annotateurs bien moins bonnes que dans le cas séparable, mais non trop aberrantes, puisqu'au terme de l'apprentissage on a : $\alpha = (0.30, 0.45, 0.50, 0.70, 0.50)$ et

$\beta = (0.50, 0.60, 0.70, 0.85, 0.40)$

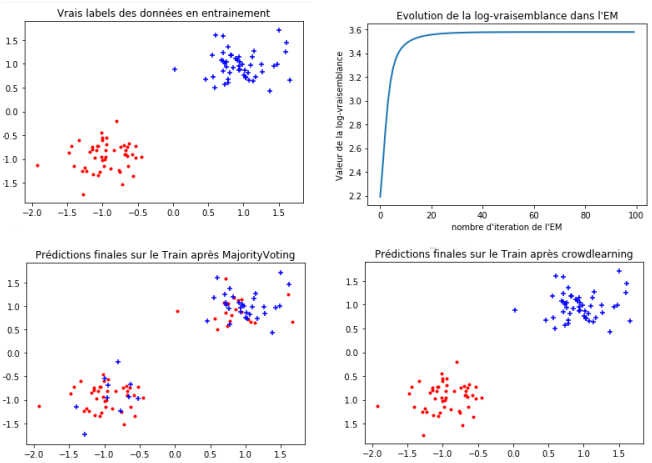

FIGURE 1 – Prédictions du premier modèle de *Crowdlearning* (non spécialisé) comparées à celles du *Majority Voting* sur l'ensemble d'apprentissage de taille $N = 100$ de données artificielles 2D du jeu 1 dans le cas séparable (bruit sur les gaussiennes : $\epsilon = 0.6$) et pour $T = 5$ annotateurs de Bernoulli de fiabilité 0.6 (sensibilité égale à la spécificité égale à 0.6). On représente aussi la courbe d'évolution de la log-vraisemblance au fil des itérations de l'EM (initialisation des $\widetilde{p}_i$ avec le *Majority Voting*)

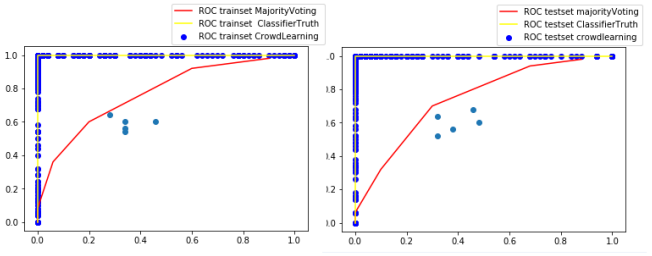

FIGURE 2 – Courbes ROC (en abscisse le taux de faux positif et en ordonnée le taux de vrai positif) pour le premier modèle de *Crowdlearning* (non spécialisé), pour le *Majority Voting* et pour le classifieur linéaire par régression logistique appris directement avec les vrais labels Z. L'ensemble d'apprentissage est celui représenté en figure 1 et l'ensemble de de test de même taille est généré avec les mêmes paramètres (cas séparable).

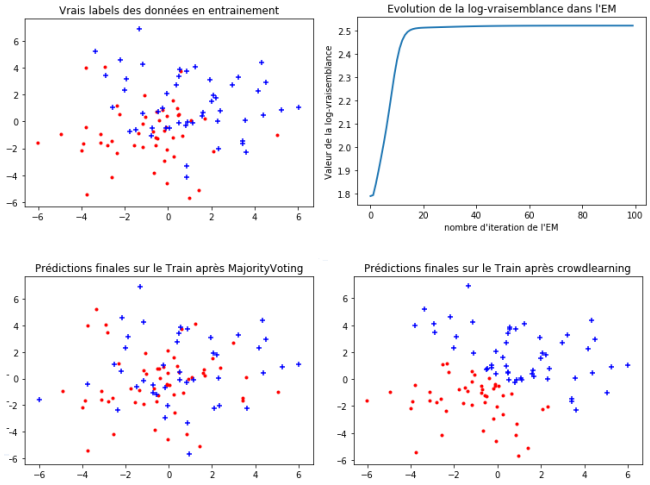

FIGURE 3 – Prédictions du premier modèle de *Crowdlearning* (non spécialisé) comparées à celles du *Majority Voting* sur l'ensemble d'apprentissage de taille $N = 100$ de données artificielles 2D du jeu 1 dans le cas non séparable (bruit sur les gaussiennes $\epsilon = 2$) et pour $T = 5$ annotateurs de Bernoulli de fiabilité 0.6 (sensibilité égale à la spécificité égale à 0.6). On représente aussi la courbe d'évolution de la log-vraisemblance au fil des itérations de l'EM (initialisation des $\widetilde{p}(z_i)$ avec le *Majority Voting*)

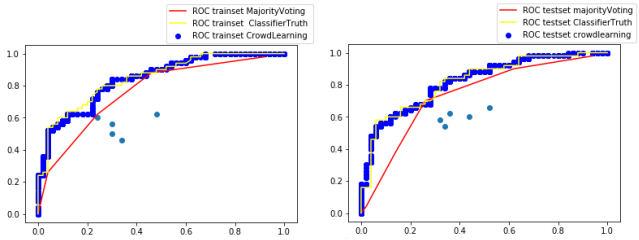

FIGURE 4 – Courbes ROC (en abscisse le taux de faux positif et en ordonnée le taux de vrai positif) pour le premier modèle de *Crowdlearning* (non spécialisé), pour le *Majority Voting* et pour le classifieur linéaire par régression logistique appris directement avec les vrais labels Z. L'ensemble d'apprentissage est celui représenté en figure 3 et l'ensemble de de test de même taille est généré avec les mêmes paramètres (cas non séparable).

Expérience 2 - On peut aussi chercher à déterminer à partir de quelle qualité d'annotation il est possible de retrouver les vrais labels avec le premier modèle de *Crowdlearning*. En figure 5, on trace aussi les performances obtenues pour ce modèle sur le même jeu des données, cas séparable, en fonction de la qualité d'une population d'annotateurs de Bernoulli (ayant tous même qualité). On observe ainsi qu'à partir d'une qualité 0.53

(valeur de sensibilité/spécificité) le *Crowdlearning* bat le jugement majoritaire et que dès que les annotateurs ont une probabilité de 0.6 de donner le bon label, on atteint les performances du classifieur parfait appris avec les vrais labels.

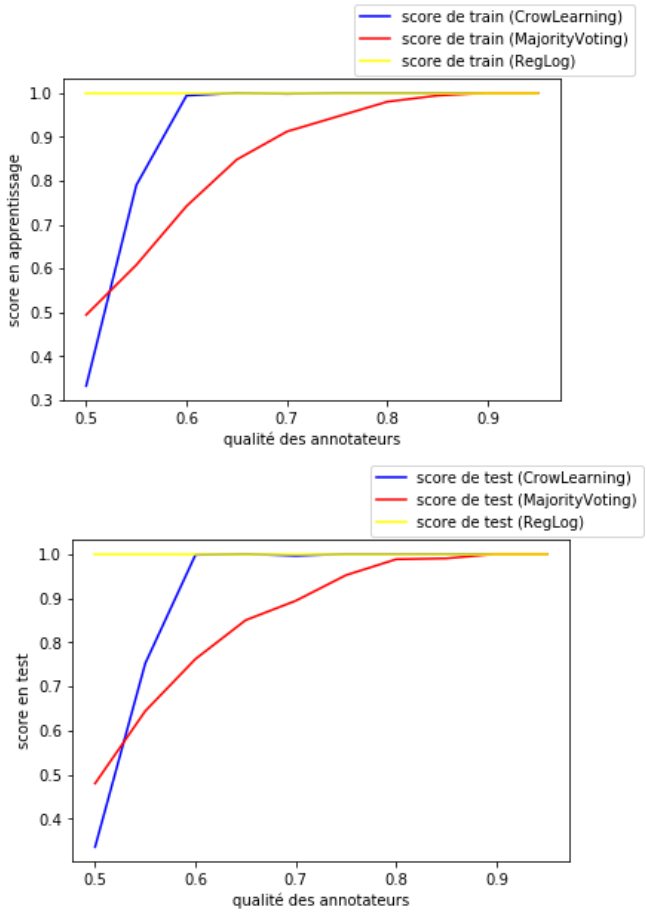

FIGURE 5 – Courbes pour le premier modèle (non spécialisé) donnant les scores de train et de test (pour un seuil de 0.5) sur le jeu 1 de données artificielles (cas séparable, $\epsilon = 0.1$) pour une population de $T = 10$ annotateurs de Bernoulli de qualité identique (en abscisses, sensibilité et spécificité égales à la qualité) . Ensembles d'apprentissage et de test de $N = 100$ données chacun.

Expérience 3 - Comme la dénomination du modèle de *Crowdlearning* laisse supposer d'un cas d'utilisation, on peut également vouloir étudier l'influence du nombre d'annotateurs dans une population sur les performances du modèle. Nous avons donc tracé l'évolution du score en train et en test en augmentant le nombre d'annotateurs, à la fois du modèle non spécialisé, du modèle *majority voting*, et du modèle de régression logistique. Afin d'obtenir des résultats plus intéressant, nous simulons 50 fois et moyennons les résultats , en tirant à chaque fois des qualités d'annotateurs entre 0.4 et 0.7 , et ce afin de simuler une population d'amateurs, parfois un peu mauvais, parfois relativement bon. Comme nous pouvons le voir dans la figure 6, le modèle non spécialisé domine très clairement le *majority voting*, atteignant des scores très satisfaisant (plus de 0.9 dès qu'on dépasse la dizaine d'annotateurs), alors que le *majority voting* reste inférieur à 0.7 sur toute la plage d'annotateurs. On voit donc bien ici l'intérêt d'un tel modèle : en multipliant les sources d'information données par des annotateurs amateurs mais en nombre, on parvient à en tirer de bien meilleurs informations qu'avec un simple *majority voting*, avec un score pour le modèle non spécialisé presque parfait lorsque le nombre d'annotateurs devient important. L'augmentation du nombre de sources d'informations permet également une augmentation des scores atteints pour les deux modèles, ce qui est conforme à l'intuition.

4.4 Résultats du second modèle (spécialisé) sur données artificielles

Expérience 4 - Attachons-nous maintenant à tester le second modèle, et plus particulièrement sur une situation où il peut donner théoriquement de meilleurs résultats que le premier modèle. Ainsi nous considérons le jeu 2 de données artificielles 2D, et deux annotateurs, le premier spécialisé dans le domaine $|x| > |y|$, le second spécialisé dans l'autre domaine (fiables à 0.9 dans leurs domaines de spécialisation respectifs, et à 0.6 dans les complémentaires respectifs de ces domaines) (cf. figure 7). Comme on l'observe en figure 8, le modèle spécialisé permet d'obtenir des prédictions toujours bien meilleures que le *Majority Voting* et sensiblement meilleures au premier modèle non spécialisé. On note en effet qu'au terme de l'apprentissage du modèle spécialisé (100 itérations de l'EM au bout desquelles on a bien convergence comme on l'observe sur la courbe), les fonctions de qualités des annotateurs (probabilités de succès d'annotation) s'écrivent respectivement :

$$\eta_1(x_i) \simeq \sigma(21.8x_{i1} + 7.8x_{i2} + 32.4)$$
$$\eta_2(x_i) \simeq \sigma(1.7x_{i1} + 7.8x_{i2} + 18.0)$$

(où σ est pour rappel la fonction sigmoïde, et où x_{i1} et x_{i2} correspondent aux abscisses et ordonnées de la donnée i), ce qui indique que le modèle a bien retrouvé les domaines de spécialisation des annotateurs : l'annotateur 1 (spécialisé pour rappel dans le domaine $|x| > |y|$) apparaît bien plus sensible aux abscisses, quand l'annotateur 2 (spécialisé pour $|y| > |x|$) l'est plus aux ordonnées. C'est la contribution de leurs deux domaines d'ex-

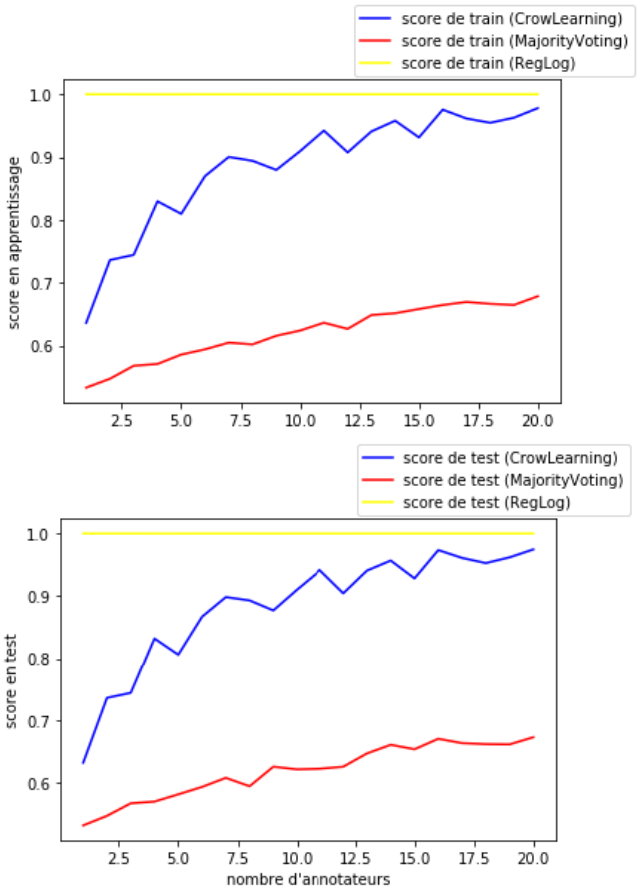

FIGURE 6 – Courbe pour le premier modèle (non spécialisé) donnant les scores de train et de test (pour un seuil de 0.5) sur le jeu 1 de données artificielles (cas séparable, $\epsilon = 0.1$, $N = 100$). On fait varier la population d'annotateurs de 1 à 20. Pour un nombre d'annotateurs fixé, on va tirer 50 fois des qualités d'annotations variant entre 0.4 et 0.7 (simulation d'une population d'annotateurs amateurs) et on moyenne les scores en train et en test obtenus sur les 50 expériences

pertises qui couvrent tout l'espace qui permet donc de retrouver la bonne frontière de classification $y = -x$, ce à quoi réussit moins bien le premier modèle non spécialisé. Ce dont témoignent les courbes ROC en apprentissage et en test obtenues pour la même expérience (figure 9) : la courbe ROC du modèle spécialisé se confond en effet dans ce cas non séparable avec le classifieur appris directement sur les vrais labels, et reste toujours strictement au dessus de la ROC du premier modèle.

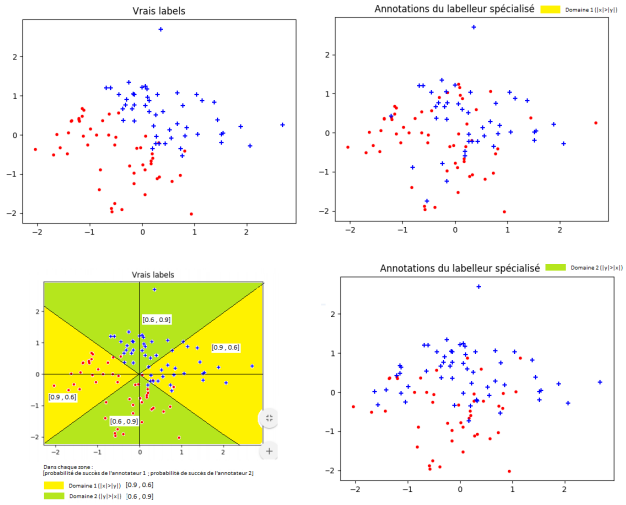

FIGURE 7 – Jeu 2 de données artificielles 2D dans le cas non séparable ($N = 100$, $\epsilon = 0.4$). Labels attribués par $T = 2$ annotateurs : l'un spécialisé dans le domaine |y|>|x| (vert), l'autre dans le domaine $|x| > |y|$ (jaune). La probabilité de succès d'un annotateur dans son domaine de spécialisation est de 0.9, et est de 0.6. dans le reste de l'espace

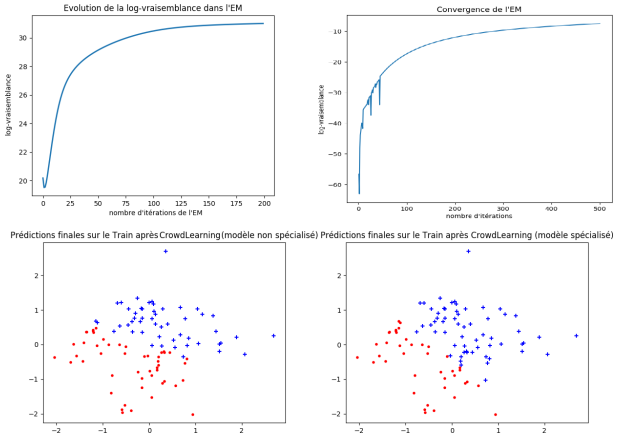

FIGURE 8 – Courbes d'évolution de la log-vraisemblance au fil des itérations de l'algorithme EM pour le modèle de *Crowdlearning* non spécialisé vue en partie 1 (à gauche) et pour le modèle de *Crowdlearning* spécialisé vu en partie 2 (à droite). Prédictions obtenues pour chacun des deux modèles sur l'ensemble d'apprentissage au terme des 200 itérations de l'EM, avec un seuil $s = 0.5$.

Expérience 5 - Aussi se pose une question assez naturelle avec le modèle spécialisé en main : obtient-on

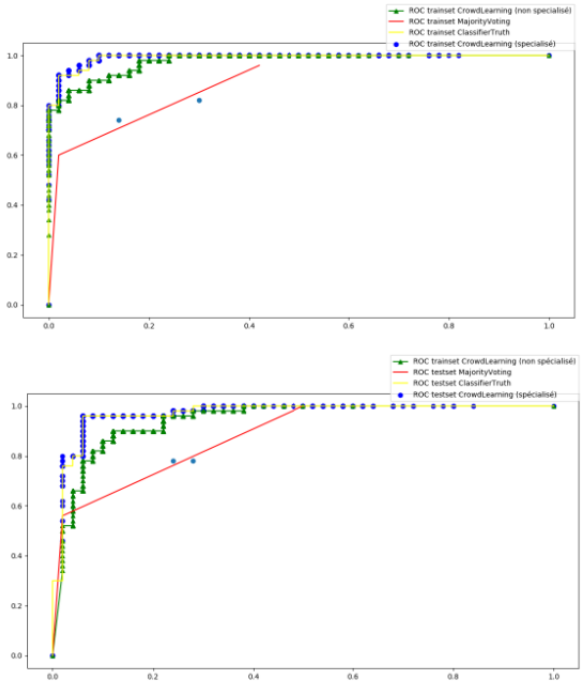

FIGURE 9 – Courbes ROC (en abscisse le taux de faux positif et en ordonnée le taux de vrai positif) pour le premier modèle de *Crowdlearning* (non spécialisé), pour le deuxième modèle de *Crowdlearning* (spécialisé) pour le *Majority Voting* et pour le classifieur linéaire par régression logistique appris directement avec les vrais labels Z. L'ensemble d'apprentissage est celui représenté en figure 7 et l'ensemble de de test de même taille est généré avec les mêmes paramètres.

sensiblement de meilleures performances avec deux annotateurs moyennement fiables sur l'ensemble des données, ou avec deux annotateurs de niveaux de connaissances très hétérogènes mais complémentaires dans deux régions distinctes ? Ainsi on reprend le second jeu de données artificielles et on considère toujours deux annotateurs, respectivement fiables à $0.5+\alpha$ et $1-\alpha$ dans le domaine $|x| > |y|$, et à $1-\alpha$ et $0.5+\alpha$ dans le domaine $|y| > |x|$. Comme on l'observe sur la figure 10, on obtient bien des courbes de scores de tests des deux modèles de *Crowdlearning* qui sont symétriques par rapport à $\alpha = 0.25$, ce qui est cohérent puisque la différence de probabilités de succès entre les deux annotateurs dans une même zone s'exprime comme 0.5-2α. On observe que le modèle non spécialisé est bien entendu d'autant moins performant que cette différence de spécialisation est importante, contrairement au modèle spécialisé qui réussit relativement mieux la classification quand les annotateurs possèdent chacun leurs vrais domaines d'expertise (avec une différence de spécialisation importante).

FIGURE 10 – Courbes donnant les scores de test (pour un seuil de 0.5) obtenus avec les deux modèles de *Crowdlearning* (non spécialisé et spécialisé), pour le *Majority Voting* et pour le classifieur linéaire par régression logistique appris directement avec les vrais labels. Sur le deuxième jeu de données artificielles (non séparable, $\epsilon = 0.4$) pour $T = 2$ annotateurs de Bernoulli, le premier spécialisé dans la zone |x|>|y|, le second dans la zone |y|>|x|. $0.5+\alpha$ correspond à la probabilité de bon label d'un annotateur dans son domaine de spécialisation, et $1-\alpha$ à cette même probabilité hors de son domaine de spécialisation.
Note importante : Les courbes de scores en test sont à analyser séparément pour les 4 modèles. En effet, le niveau global d'une courbe par rapport à l'autre ne signifie rien étant donné que 0.5 est un meilleur seuil pour certains modèles alors que d'autres seraient plus performants avec un autre seuil. On a déjà discuté de la différence de performances entre les 4 modèles avec les courbes ROC ; on s'intéresse ici à leur qualité relativement à α. Ensembles d'apprentissage et de test de $N = 100$ chacun.

4.5 Résultats des deux modèles sur données réelles

Il est temps maintenant d'appliquer les modèles non spécialisé et spécialisé aux données réelles de l'*Adult dataset* de UCI Machine Learning.

Nous testons les performances des deux modèles avec un *slicing* de 80% en données d'entraînement pour 20%

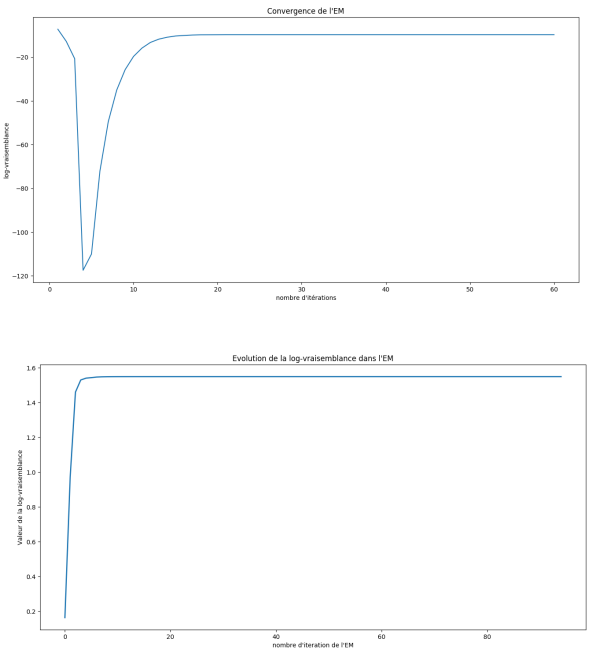

FIGURE 11 – Courbes de convergence des algorithmes EM sur le jeu d'entrainement des données réelles.
On peut voir ici la convergence pour nos deux modèles (spécialisé en haut et non spécialisé en bas)

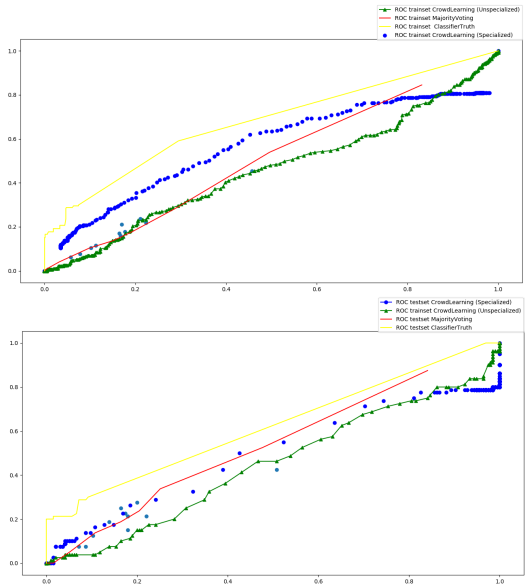

FIGURE 12 – Courbes ROC en Train et en Test sur les données réelles pour le deux modèles de *Crowdlearning* (non spécialisé et spécialisé) pour le *Majority Voting* et pour le classifieur linéaire par régression logistique appris directement avec les vrais labels.

de données en test de l'ensemble de départ. Sur les courbes ROC, on observe que les performances du modèle non spécialisé sont très médiocres, avec une courbe ROC toujours en dessous de la ROC du *Majority Voting* à la fois en entraînement et en test. Le modèle spécialisé lui, montre sa supériorité sur le *Majority Voting* au moins sur la courbe d'entraînement, donnant ainsi des espoirs en le modèle, même s'il faut bien reconnaître que ses prédictions ne demeurent que légèrement meilleures que celles d'un classifieur aléatoire. La supériorité du modèle 2 sur le modèle 1 semble en tous cas indiquer qu'il existe des domaines d'expertises pour chaque annotateur dans notre espace des données à 13 dimensions. Enfin la ROC de la régression logistique reste au-dessus de toutes les autres à la fois en entraînement et en test, ce qui est attendu puisque nous faisons l'optimisation des poids dans ce classifieur en connaissance des vrais labels et que nous recherchons une relation entre X et Z de la même forme dans les deux modèles de *Crowdlearning*.

Enfin, les différences de qualité des ROC du modèle spécialisé entre entraînement et test semblent témoigner d'un certain un phénomène de sur-apprentissage même si le nombre de dimensions descriptives (caractéristiques socio-économiques) demeure restreint ($d = 13$). Cela nous a alors conduit à adopter une approche de *ridge regression* (en ajoutant un terme de régularisation - $\lambda * ||w||^2$ à l'opposé de la vraisemblance, fonction objectif à minimiser) pour améliorer les résultats du modèle spécialisé. Cependant, la recherche du coefficient λ optimal pour la *ridge regression* n'a pas été concluante.

Pour expliquer les performances assez décevantes par rapport aux données artificielles des modèles, on peut avancer plusieurs points :

— Tout d'abord, les résultats que nous obtenons nous donnent que sur 11 annotateurs, 6 possèdent une spécificité très faible (très inférieur à 0.5) et une autre moitié possède une spécificité nettement supérieure à 0.5. D'autre part, tous les annotateurs sont accrédités d'une sensitivité très élevée (supérieure à 0.8). Premièrement, il est donc assez difficile pour l'algorithme de savoir lequel des deux groupe est meilleur (symétrie par rapport à 0.5). Aussi ces résultats semblent également indiquer une certaine spécialisation des annotateurs, ce qui est assez cohérent avec le fait que le classifieur spécialisé est meilleur que le *Majority Voting* et le classifieur non spécialisé.

— Ensuite, les hypothèses des modèles sont certainement à remettre en cause également. Si on suppose

que le niveau de connaissance des labelleurs varie selon le lieu de la donnée, la fonction de connaissance des annotateurs η ne s'exprime probablement pas de manière linéaire en fonction des *features*.

— Enfin, le modèle n'est sans doute pas assez expressif. Tenter de séparer par une droite les données (salaires à $+$ ou $-$ $\$50k$) dans cet espace à 13 dimensions (les caractéristiques socio-économiques) est ici illusoire car comme nous le montre la ROC du classifieur RegLog, nous sommes très loin des résultats qu'on devrait obtenir avec un classifieur parfait. Pour prédire le label d'une donnée avec un classifieur linéaire, il faut peut-être rajouter d'autres dimensions à notre espace, soit à l'aide d'autres caractéristiques, soit à partir de ces 13 caractéristiques en les couplant entre elles, par exemple via un plongement polynomial ou gaussien.

5 *Bonus* - Esquisse d'un dernier modèle "dépendant" où une consigne extérieure peut orchestrer certains annotateurs

5.1 Description du modèle

Jusque là on a supposé l'indépendance des labels posés par les annotateurs conditionnellement à la donnée et au vrai label. En pratique, on comprend qu'il peut exister des corrélations entre les annotateurs indépendamment de la donnée, pouvant se manifester par exemple par la présence d'une croyance populaire, ou d'une consigne de vote qui dicte un label connu de tous, et auquel les annotateurs sont plus ou moins sensibles, pouvant donc y adhérer ou s'y opposer.
On comprend bien alors l'intérêt qu'il peut y avoir à repérer ces individus influençables ou réactionnaires à la croyance populaire ou à la consigne de vote, pour ne plus tenir compte de leurs annotations dans la prédiction, et pour accorder du poids uniquement aux individus tentant de vraiment prédire en fonction de la donnée.

Pour cette section, on reprend le cadre du modèle à annotateurs spécialisés. On suppose ensuite qu'il existe un organisme de propagande connue de tous demandant aux annotateurs d'attribuer le label 1 quelle que soit la donnée. On suppose que chaque labelleur t est plus ou moins sensible à la propagande. Sa sensibilité est caractérisée par $\nu_t \in [0,1]$ sa propension à voter en fonction de la consigne de propagande plutôt que donner un label en puisant dans sa connaissance vis-à-vis de la donnée, et par $s_t \in [0,1]$ sa propension à aligner son label sur le label 1 de propagande s'il décide de voter finalement en fonction de la propagande.

Ainsi, avec la formule des probabilités totales, on peut réécrire :

$$\begin{aligned}\mathbb{P}(y_i^t|z_i,x_i) = \quad & (1-\nu_t))\eta_t(x_i)^{|y_i^t - z_i|} \\ & (1-\eta_t(x_i))^{1-|y_i^t-z_i|} \\ & +\nu_t s_t^{y_t}(1-s_t)^{1-y_t}\end{aligned}$$

Ainsi, même si on a introduit une certaine dépendance des annotateurs vis-à-vis d'une consigne extérieure, on garde une indépendance conditionnelle des Y_i^t sachant la donnée, le vrai label, et les sensibilités ν_t, s_t (et sachant le fait que la consigne dit de voter le label 1). Ainsi, on peut toujours écrire la log-vraisemblance du modèle sous la forme :

$$\theta \to L(X,Y|\theta) = \prod_{i=1}^{N}\prod_{t=1}^{T}\mathbb{P}(y_i^t|x_i,\theta)$$

notant $\theta = \{\alpha_t, \beta_t, w, \gamma, \nu_t, s_t\}$

On peut à nouveau utiliser un algorithme EM pour la maximiser, avec un algorithme de type descente de gradient pour l'étape de maximisation puisqu'il n'existe pas de forme close pour les meilleurs paramètres comme pour le modèle spécialisé.

5.2 Expérience de test du modèle et résultats

Pour tester le modèle, on peut reprendre le premier jeu de données artificielles en 2D et considérer par exemple trois groupes d'annotateurs : deux groupes sensibles à la consigne de vote, l'un ayant une forte inclinaison à la suivre et l'autre ayant une forte inclinaison à y réagir en donnant son opposé quelle que soit la donnée, puis un troisième groupe insensible à la consigne n'utilisant donc que sa connaissance vis-à-vis de la donnée pour prédire le label, et ayant une probabilité de succès uniforme.

Nous avons alors cherché à savoir si le modèle était capable de distinguer ce qui dans l'attribution de chaque label des annotateurs est lié à leur connaissance vis-à-vis de la donnée, ou à leur sensibilité vis-à-vis de la consigne extérieure qui corrèle en un sens leurs décisions indépendamment de la donnée. Autrement dit, ce qui nous intéressait essentiellement à l'issue de l'apprentissage étaient les valeurs de α et β trouvés, caractéristiques des connaissances propres des annotateurs, et les ν et s caractéristiques de leurs sensibilités à la consigne. À l'issue de l'apprentissage, l'idée était ainsi ne plus tenir compte des annotations ayant de fortes valeurs de ν

pour relancer un des deux premiers modèles en ne considérant pour l'ensemble d'apprentissage que les labels des annotateurs peu sensibles à la consigne.

Nous avons donc implémenté ce modèle et nous avons expérimenté ce test avec 10 annotateurs. Toutefois nous n'avons pas pu retrouver les paramètres α, β, ν, s des annotateurs pour lesquels nous avions généré leurs labels. Premièrement, les valeurs de ν et s obtenus en fin d'apprentissage sortaient largement de l'intervalle $[0, 1]$. Ainsi une première correction simple au modèle aurait pu consister à modifier l'hypothèse du modèle comme suit :

$$\begin{aligned}\mathbb{P}(y_i^t|z_i, x_i) = \quad & (1-\sigma(\nu_t))\eta_t(x_i)^{|y_i^t - z_i|} \\ & (1-\eta_t(x_i))^{1-|y_i^t - z_i|} \\ & +\sigma(\nu_t)\sigma(s_t)^{y_t}(1-\sigma(s_t))^{1-y_t}\end{aligned}$$

(la fonction sigmoïde permettant de ramener tout paramètre réel dans $[0, 1]$)

Par ailleurs il nous est apparu après cette expérience que, au-delà de ce premier souci, ce modèle présente certainement un trop grand nombre de degrés de liberté. Autrement dit, on peut craindre que le modèle ait du mal à identifier ce qui dans un label erroné d'un labelleur provient de sa mauvaise connaissance (η) ou de sa sensibilité à la consigne de vote ou à la croyance populaire (ν, s) non reliée à la donnée. Ainsi, l'étude implicite des corrélations entre les labels des différents annotateurs pourrait ne pas suffire à deviner l'existence de cette consigne tant les sensibilités des annotateurs sont différentes.

6 Conclusion

Si l'apprentissage supervisé repose traditionnellement sur un seul annotateur, la difficulté d'obtenir la vérité terrain pour certaines données ou le fait que celle-ci soit tout simplement plus subjective montre la nécessité de développer des méthodes de classification à partir de plusieurs étiquetage bruités, qui de surcroît sont de plus en plus accessibles avec des phénomènes tels que l'OpenSource.

Dans ce projet, nous avons implémenté deux modèles de classification établis par deux articles : le premier s'attachant à distinguer sensitivité et spécificité pour chaque annotateur, le second faisant varier le niveau de connaissance de chaque annotateur en fonction de la région où se trouve la donnée. Les expériences effectuées sur notre jeu de données socio-économiques semblent montrer l'intérêt particulier de ces domaines d'expertises des annotateurs, permettant d'estimer plus finement et de rassembler les savoirs de chacun. Le troisième modèle que nous avons introduit, où les annotateurs peuvent être orchestrés ou non par une consigne extérieure ou une croyance populaire par exemple, rajoutant ainsi une certaine dépendance entre les annotateurs, n'a pas fait ses preuves sur les données artificielles ; nous attribuons en partie cet échec aux degrés de libertés supplémentaires introduits avec les sensibilités de chacun vis-à-vis de la consigne. Toutefois, nous pensons qu'en rajoutant d'autres hypothèses plus contraignantes sur la forme des niveaux de connaissance des annotateurs et de la dépendance à la consigne, le problème pourrait être surmonté et le modèle pourrait ainsi trouver son utilité pour détecter à la fois les niveaux de connaissance et leurs sensibilité à une consigne populaire. Par ailleurs, nous aurions aimé tester les performances des deux premiers modèles sur des données médicales (diagnostics de spécialistes pour des cancers par exemple) et étendre notre travail aux problèmes multi-classes (détermination du type de maladies lorsque plusieurs d'entre elles présentent des symptômes similaires par exemple). De tels données ont manquées, mais sous la condition que les médecins n'aient pas des spécificités ou des sensibilités trop inférieures à 0.5 (notre exemple économique a pu exhiber les problèmes liés à de telles situations), nous pouvons croire en l'efficacité du *Crowdlearning* après les diverses expériences menées à bien sur les données artificielles.

Remerciements

Nous tenons tout d'abord à remercier nos encadrants, M. Guillaume Obozinski, M. Nicolas Baskiotis et M. Benjamin Dubois pour le temps qu'ils nous ont accordé et pour la pertinence de leurs conseils. Les échanges que nous avons pu avoir avec eux ont été très précieux pour l'avancement et la réussite de notre projet.

Bibliographie

[1] Vikas C. Raykar, Shipeng Yu, Linda H. Zhao, Gerardo Hermosillo Valadez, Charles Florin, Luca Bogoni, Linda Moy, Learning From Crowds, Journal of Machine Learning Research 11, pp. 1297-1322, 2010.

[2] Yan Yan, R'omer Rosales, Glenn Fung, Mark Schmidt, Gerardo Hermosillo, Luca Bogoni, Linda Moy, and Jennifer Dy Modeling Annotator Expertise : Learning when Everybody Knows a Bit of Something, In Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), 2010.

[3] Outils de génération de données artificielles des TP du cours de Machine Learning ENPC 2ème année 2017

[4] Données : UCI Machine Learning, "Adult dataset"

Lien GitHub - `https://github.com/AmineKheldouni/Machine-Learning-Project`

École des Ponts

ParisTech

Appendice

Après l'analyse des articles proposés, nous avons jugé bon de reprendre les calculs des gradients selon les différents paramètres de nos vraisemblances (notamment car il manquait certaines constantes multiplicatives dans l'article [2] et car notre implémentation en pâtissait ainsi). Les résultats des calculs sont présentés ci-dessous pour les différents modèles présentés dans le rapport.

Formules du gradient et de la Hessienne de la vraisemblance espérée du premier modèle de *Crowdlearning* (non spécialisé)

La formule du gradient en W de la fonction de vraisemblance espérée du premier modèle s'écrit comme suit :

$$g(W) = \sum_{i=1}^{N} (\sigma(W^T X_i) - \widetilde{p}_i) x_i$$

avec N le nombre d'exemple, et X_i l'exemple i.

En dérivant à nouveau, on aboutit à la formule de la Hessienne en w de la fonction de vraisemblance espéré, s'écrivant :

$$H(W) = \sum_{i=1}^{N} (\sigma(W^T X_i))(1 - \sigma(W^T X_i)) X i X_i^T$$

Précisons que comme nous cherchons à maximiser la fonction de vraisemblance, nous utilisons les opposés des gradients et Hessiennes calculées précédemment dans nos algorithmes pour optimiser W par descente de gradient.

Formules du gradient de la vraisemblance espérée du second modèle de *Crowdlearning* (spécialisé)

Les formules de dérivation par rapport aux différentes variables $\theta = (w,\ \gamma,\ \alpha_t,\ \beta_t)$ de la vraisemblance espérée du second modèle de *Crowdlearning* (spécialisé), nécessaires à la descente de gradient à pas variable, sont données dans cette section.

D'une part, les gradients en α_t et β_t pour $1 \leq t \leq T$ sont données par :

$$\begin{aligned} \nabla_{\alpha_t} l(W, X, Y) = \quad & \sum_{i=1}^{N} X_i \sigma(\alpha_t T X + \beta_t)(\widetilde{p}(z_i)[(1 - |y_t^i - 1|) \\ & \exp(-\alpha_t T X - \beta_t) - |y_t^i - 1|] + (1 - \widetilde{p}(z_i)) \\ & [(1 - |y_t^i|) \exp(-\alpha_t T X - \beta_t) - |y_t^i|] \end{aligned}$$

$$\nabla_{\beta_t} l(W,X,Y) = \sum_{i=1}^{N} \sigma(\alpha_t TX + \beta_t)(\widetilde{p}(z_i)[(1-|y_t^i - 1|) \exp(-\alpha_t TX - \beta_t) - |y_t^i - 1|] + (1-\widetilde{p}(z_i)) [(1-|y_t^i|)\exp(-\alpha_t TX - \beta_t) - |y_t^i|]$$

D'autre part, Les gradients en w et γ s'écrivent sous la forme suivante :

$$\nabla_w l(W,X,Y) = T\sum_{i=1}^{N} X_i \sigma(\alpha_t TX + \beta_t)(\widetilde{p}(z_i)\exp(-\alpha TX - \beta) - (1-\widetilde{p}(z_i))$$

$$\nabla_\gamma l(W,X,Y) = T\sum_{i=1}^{N} \sigma(\alpha_t TX + \beta_t)(\widetilde{p}(z_i)\exp(-\alpha TX - \beta) - (1-\widetilde{p}(z_i))$$

Formules de gradients pour le classifieur linéaire par régression logistique simple

Pour le classifieur "RegLog" appris sur la vérité terrain Z, le calcul du gradient de la fonction de coût en W (opposé de la vraisemblance) a abouti à la formule qui suit :

$$\begin{aligned} \nabla_W - l(W,X,Z) &= \sum_{i=1}^{N} \frac{\exp -Z^i \widetilde{X}^i W}{1+\exp -Z^i \widetilde{X}^i W} - Z^i \widetilde{X}^{iT} \\ &= \sum_{i=1}^{N} \left(\frac{-Z^i \exp -Z^i \widetilde{X}^i W}{1+\exp -Z^i \widetilde{X}^i W} \right) \widetilde{X}^{iT} \end{aligned}$$

notant $\widetilde{X}^i$ le vecteur constitué par la ligne i de la matrice $\widetilde{X}$