THÈSE

En vue de l'obtention du

DOCTORAT

Délivré par : *L'Université d'Aix-Marseille*

Présentée et soutenue le *01/12/2014* **par :**

CHIRINE WOLLEY

APPRENTISSAGE SUPERVISE A PARTIR DE MULTIPLES ANNOTATEURS INCERTAINS

JURY

THIERRY ARTIÈRE	Professeur, Université d'Aix-Marseille	Examinateur
YOUNES BENNANI	Professeur, Université Paris Nord	Rapporteur
FAICEL CHAMROUKHI	Maitre de Conférences, Université de Toulon	Examinateur
PHILIPPE LERAY	Professeur, Université de Nantes	Rapporteur
MOHAMED QUAFAFOU	Professeur, Université d'Aix-Marseille	Directeur de Thèse
JEAN SALLANTIN	Directeur de Recherche, Université de Montpellier	Examinateur

École doctorale et spécialité :

Ecole Doctorale en Mathématiques et Informatiques de Marseille

Unité de Recherche :

Laboratoire des Sciences de l'Information et des Systèmes (LSIS)

A la mémoire de mon grand-père.

Résumé

En apprentissage automatique supervisé, obtenir les réels labels pour un ensemble de données peut s'avérer être une étape très fastidieuse et très longue. De plus, les données collectées peuvent être contaminées d'erreurs, caractérisées par la présence de valeurs manquantes, etc. Ainsi, de nombreuses méthodes ont été développées pour faire face à l'imperfection aussi bien des données de description d'instances que de leurs étiquettes (labels). Aujourd'hui, les récentes avancées d'Internet ont permis le développement de nombreux services d'annotations en ligne, faisant alors appel au crowdsourcing pour collecter facilement et rapidement des labels. Néanmoins, le principal inconvénient de ces services réside dans le fait que les annotateurs peuvent avoir des niveaux d'expertise très hétérogènes. Ainsi, le spectre des annotateurs varie des experts, peu nombreux, jusqu'aux incompétents, en passant par une majorité d'annotateurs qui n'ont qu'une connaissance partielle du problème. De telles données ne sont forcément pas fiables, de plus quelques annotateurs peuvent même être des spammers. Par conséquent, la gestion de l'incertitude des annotateurs est un élément clé pour l'apprentissage à partir de multiples annotateurs de niveaux de compétences très hétérogènes.

Dans cette thèse, nous proposons des algorithmes probabilistes qui traitent l'incertitude des annotateurs et la qualité des données durant la phase d'apprentissage. Pour cela, nous donnons la possibilité aux annotateurs d'exprimer leur incertitude durant le processus de labellisation. On se restreint aux deux cas suivants : (1) Ignorance totale, l'annotateur utilise le symbole « ? » lorsqu'il ne peut quantifier son incertitude et (2) Ignorance partielle, l'annotateur est capable d'expliciter un degré de certitude associé au label qu'il propose. Les trois modèles proposés dans cette thèse sont des modèles d'apprentissage en présence de multiples annotateurs incertains, et permettent de classer une nouvelle instance tout en réalisant une tâche additionnelle particulière. Ces modèles adoptent une approche probabiliste Bayésienne et se basent sur les modèles graphiques et des méthodes d'optimisation telles que Newton-Raphson et LBFGS quasi Newton. Le modèle IGNORE permet de classer de nouvelles instances tout en évaluant les annotateurs en terme de performance d'annotation qui dépend de leur incertitude. Il devient alors possible d'exhiber un classement des annotateurs. Le modèle, X-IGNORE, intègre la qualité des données en plus de l'incertitude des juges. En effet, X-IGNORE suppose que la performance des annotateurs dépend non seulement de leur incertitude mais aussi de la qualité des données qu'ils annotent. Par ailleurs, ce modèle permet d'évaluer la qualité des données en terme de leur difficulté à être annotées, et permet de prédire la

qualité d'une nouvelle instance. Enfin, le modèle ExpertS répond au problème de sélection d'annotateurs durant l'apprentissage. ExpertS élimine les annotateurs les moins performants, et se base ainsi uniquement sur les labels des bons annotateurs (experts) lors de l'étape d'apprentissage. De nombreuses expérimentations, effectuées sur des données synthétiques, montrent la performance et la stabilité de nos modèles par rapport à différents algorithmes de la littérature. Nous avons aussi exploité nos modèles dans une application médicale réelle qui consiste en la reconnaissance du mélanome à partir d'images annotées par de multiples dermatologues.

Remerciements

Me voilà, m'apprêtant à écrire les dernières lignes de cette thèse. Cette fois c'est bien le point final. La fin de trois années de thèse, cette thèse qui occupait continuellement mes pensées. Je lisais souvent les remerciements d'autres thésards, en essayant d'imaginer l'agréable ressenti qu'ils pouvaient avoir. Je me rends aujourd'hui compte que ce ressenti que j'imaginais était loin de la réalité. En effet, seuls les sentiments de soulagement et de fierté me venaient à l'esprit. Mais écrire ces derniers mots de ma thèse provoque en réalité des sentiments bien plus contrastés. De ces sentiments de joie découlent des sentiments de nostalgie, des sentiments de fierté découlent des sentiments de gratitude envers tous ceux qui m'ont aidé. Ces trois années de thèse n'auront en effet jamais pu aboutir sans mon entourage qui a su m'aider à garder le cap dans les moments difficiles pour atteindre l'objectif tant désiré. Il m'a paru donc essentiel de consacrer cette page à remercier chacune de ces personnes ayant participées à l'aboutissement de ce travail.

Un grand merci tout d'abord à mon directeur de thèse, le Pr. Mohamed Quafafou, d'avoir accepté d'encadrer cette thèse, d'avoir eu confiance en moi, d'être resté patient lorsque je piétinais, et de ses encouragements dans mes moments de doute. Je le remercie également pour les nombreux conseils qu'il m'a apportés tout au long de cette thèse, et d'avoir entièrement lu et corrigé ce manuscrit. Je réalise aujourd'hui à quel point son investissement et son soutien ont été pour moi importants dans la conduite de ce travail. Je le remercie profondément pour tout.

Je remercie les Professeurs Younes Bennani et Philippe Leray pour m'avoir fait l'honneur d'accepter d'être rapporteur de ce mémoire. Je les remercie pour leur remarques et commentaires aussi bien sur la forme que sur le fond de mon travail. Je remercie également Mr. Thierry Artière, Mr. Jean Sallantin et Mr. Faicel Chamroukhi pour leur participation à mon jury de thèse.

Je tiens également à remercier Dr. Gilles Nachouki, sans qui cette thèse n'aurait jamais eu lieu, puisque c'est grâce à lui que j'ai eu l'opportunité d'effectuer ces trois années de doctorat sous la direction du Pr. Mohamed Quafafou.

Un immense merci à l'ensemble du laboratoire LSIS, avec qui j'ai partagé d'agréables moments durant ces trois années de thèse. Mes pensées vont particulièrement à ma voi-

sine de bureau Sana et à Rabbah. Je ne cesserais de me rappeler les nombreux fous rires que l'on a eus ensemble. Je n'oublie bien entendu pas tous les autres membres du laboratoire (Shereen, Remiel, Lama, Chahinez, Radhia, Talal, ...). Je vous souhaite à tous une très bonne continuation et beaucoup de courage à tous ceux qui sont encore en train de galérer en thèse ;)

Mes sentiments les plus chers vont bien évidemment à toute ma famille, mon frère, ma soeur, et plus particulièrement à mes parents et mes grands-parents. Sans votre soutien, je n'aurai probablement jamais pu finir ce travail. Vous avez vécu ces trois années de doctorat avec moi, en partageant mes moments de doute et de joie. Je sais que cela n'a pas toujours été facile pour vous. Je vous remercie infiniment pour la patience dont vous avez fait preuve dans mes moments difficiles. Cette thèse est aussi en quelque sorte la vôtre. J'ai enfin une énorme pensée à mon grand-père Ahmad Nazir Nachouki, qui nous a quitté il y a peu. J'aurai tant aimé que tu sois présent avec nous en ce jour.

Enfin, comme on dit en anglais, last but not least, je remercie profondément mon mari, qui a fait preuve de beaucoup de patience durant cette dernière année de thèse. Je te remercie de ton soutien, de ton aide, et d'avoir accepter de lire toute ma thèse...bien que tu te sois arrété à la première page de l'introduction, c'est l'intention qui compte :p. J'ai bien peur qu'il faille trouver un nouveau sujet de dispute à partir de maintenant :D Je te remercie enfin et surtout du magnifique cadeau que tu m'as fait en cette fin de thèse...notre petite fille Zéna.

Une page se tourne, une autre s'écrit...

Table des matières

Table des figures

Liste des tableaux

Chapitre 1

Introduction du Contexte et de la Problématique de Recherche

Sommaire

1.1 Apprentissage Automatique

Dans sa nature, l'Homme a toujours tendance à vouloir apprendre de ses expériences passées, de ses échecs et de ses réussites, afin de pouvoir contrôler et prédire au mieux possible sa vie future. Prenons l'exemple d'un étudiant ayant réussi un examen suite à une méthode de révision précise. Cet étudiant choisira très certainement cette même méthode pour l'examen suivant, afin de garantir au maximum sa réussite. Ainsi, chacune de nos expériences est représentée par ses caractéristiques et son contexte, qui sont emmagasinées dans notre mémoire au fur et à mesure. Ces informations sont alors utilisées pour nous aider à prendre une décision lorsque l'on rencontre une nouvelle situation

[Ormrod, 2012]. Or dans un contexte plus scientifique, cet apprentissage humain est plus connu sous le nom d'apprentissage automatique (Machine Learning en anglais).
L'apprentissage automatique a pour principal objectif d'analyser un ensemble d'observations préalablement recueillies dans le but de construire une procédure permettant de classer, d'estimer, de regrouper ou encore de prédire de nouvelles données [Mitchell, 1997]. En d'autres termes, il s'agit d'écrire et d'implémenter des algorithmes permettant, à partir d'un ensemble d'observations de départ, de créer un modèle dans le but d'exploiter les cas futurs. Le champ d'application de l'apprentissage automatique est très large. On peut le retrouver en robotique, en analyse financière, en diagnostic médical, et dans bien d'autres applications. En voici quelques exemples :

- Filtrage du courrier électronique : on souhaite écrire un programme qui filtre automatiquement le courrier reçu dans l'un des deux groupes *Email* ou *Spam*. Une analyse des précédents courriers reçus attribuera des caractéristiques spécifiques aux deux groupes, permettant ainsi de générer le classifieur [Blanzieri and Bryl, 2008].
- Détection d'un ensemble de gènes définissant une fonction biologique précise : L'utilisation d'une puce ADN permet de connaitre les niveaux d'expression des gènes dans les tissus. Le problème est alors de définir des groupes de gènes ayant la même fonction biologique [Molla et al., 2004].
- Prévision de la météo : on souhaite prévoir la température suivant un certain nombre de mesures préalablement recueillies [Sallis et al., 2011].

Le graphique 1.1 résume les plus importantes avancées sur l'apprentissage automatique durant ce dernier siècle.

Les méthodes d'apprentissage peuvent être regroupées en trois grandes catégories :

- Apprentissage supervisé : l'objectif est de construire un classifieur à partir d'un ensemble d'observations de la forme (description-classe), dans le but de classer de nouvelles observations. Ainsi, dans ce contexte, les entrées ainsi que les sorties des données initiales sont observées. L'exemple du filtrage de courrier fait partie de cette catégorie d'apprentissage. Plusieurs extensions sont associées à l'apprentissage supervisé classique. On peut citer par exemple le cas multi-labels, où les classes ne sont pas mutuellement exclusives, mais chaque instance peut appartenir à plusieurs classes simultanément [Kanj, 2013], ou encore le cas de l'apprentissage à partir de multiples annotateurs. Dans ce dernier cas, le réel label pour chaque instance du jeu de données est très difficile à obtenir et est alors remplacé par des labels provenant de multiples annotateurs non experts [Raykar et al., 2010, Yan et al., 2010].
- Apprentissage non supervisé : Seule la description est connue et pas la classe. L'algorithme développé doit alors, par lui-même, découvrir la structure des données. L'exemple de la détection de gènes appartient à cette catégorie d'apprentissage.
- Apprentissage semi-supervisé : dans le cas où certaines sorties seulement sont observées.

Pour une introduction plus approfondie sur l'apprentissage automatique, le lecteur pourra se référer aux ouvrages suivants : [Mitchell, 1997, Wasserman, 2004, Bishop, 2007, Hastie et al., 2001]. Dans le cadre de cette thèse, on s'intéresse plus particulièrement aux problèmes d'apprentissage supervisé, où chaque donnée du jeu d'apprentissage (jeu de données initial) est étiquetée par plusieurs annotateurs humains. Dans la prochaine section, on commence par définir plus précisement le cadre de l'apprentissage supervisé.

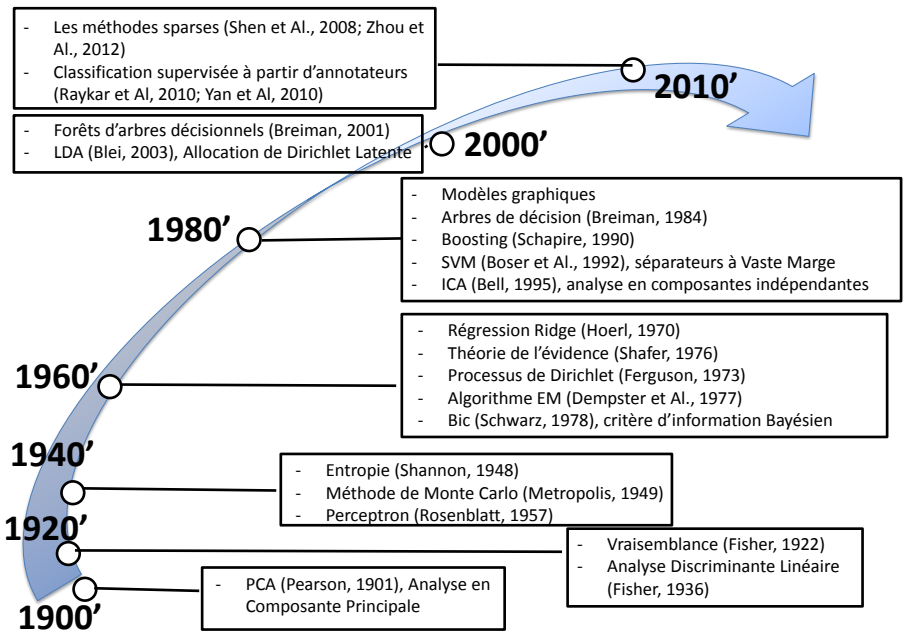


FIGURE 1.1 – Les Importantes Avancées en Apprentissage Automatique des Années 1900 à nos Jours.

1.2 Apprentissage Supervisé

1.2.1 Généralités

L'apprentissage supervisé a pour principal objectif de rechercher des fonctions permettant de prédire une variable d'intérêt, à partir d'un ensemble d'observations. Il existe trois types d'apprentissage supervisé, suivant la nature de la variable à prédire :

– Classification : Chaque instance du jeu de donnée est associée à un label. La variable d'intérêt à prédire pour une nouvelle instance est alors cette classe. Par exemple, lors d'un examen médical, on cherche à savoir si le patient est malade (classe 0) ou s'il est sain (classe 1) [Kotsiantis, 2007].

– Régression : Ici, l'instance n'est pas associée à un label discret, mais à une valeur réelle. On peut prendre l'exemple de la prédiction de la température pour un jour donné en prenant en compte différentes variables, telles que la pression de l'atmosphère, le taux de dioxyde de carbone, etc [Cornillon and Matzner-Lober, 2006].
– Séries Temporelles : Ce cas s'intéresse à la prédiction d'une valeur future, en prenant en compte ses valeurs passées. Cela correspond par exemple à un problème de bourse, où l'on souhaite prédire le taux de rendement d'une action [Bourbonnais and Terraza, 1998].

Dans cette thèse, on s'intéresse particulièrement aux problèmes de classification supervisée.

1.2.2 Classification Supervisée

La formulation mathématique de la classification supervisée est simple : Soit un jeu de données $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}$ contenant N instances x_i associée à un label y_i. x_i est un vecteur de dimension d (chaque instance x_i est décrite par d descripteurs) et y_i est catégoriel, puisque l'on se place dans un contexte de classification. Par exemple, dans un contexte de classification binaire, $y_i \in \{0, 1\}$. Soit $\mathcal{X}$ l'ensemble des données de départ, et $\mathcal{Y}$ l'ensemble des labels associés aux données. Le but est alors de générer un classifieur $f : \mathcal{X} \to \mathcal{Y}$ qui prédit la valeur y pour une nouvelle instance x donnée. De très nombreuses méthodes ont été développées afin de générer le classifieur, les plus populaires d'entre elles étant par exemple les k plus proches voisins (KNN) [Fix and Hodges, 1951] ou les Séparateurs à Vaste Marges (SVM) [Boser et al., 1992].
Le graphique 1.2 montre alors le scénario classique de la classification supervisée.

On remarque qu'une des contraintes pour l'utilisation de ces méthodes est l'obligation d'avoir recours à un expert dans le processus d'annotation du jeu de données. Cette étape peut s'avérer être très fastidieuse et très longue. Par exemple, dans le domaine médical du diagnostic assisté par ordinateur, le réel label (savoir si la région suspecte est bénigne ou pas) peut seulement être obtenu à l'aide d'une biopsie du tissu. Or la réalisation d'une biopsie est coûteuse, peut prendre beaucoup de temps et est potentiellement dangereuse pour le patient. Les récentes avancées technologiques telles que le développement d'Internet ont permis la création de nombreux sites d'annotations en ligne faisant alors appel au crowdsourcing (annotations en masse) pour répondre au problème [von Ahn and Dabbish, 2004, von Ahn et al., 2008]. C'est ce que nous allons développer dans la prochaine section.

1.3 Crowdsourcing

Dans un contexte général, le crowdsourcing [Howe, 2008] est le processus qui vise à déléguer une tache précise à un groupe de personnes, le plus souvent via un appel à tous sur internet. L'exemple le plus connu de l'utilisation du crowdsourcing est la fameuse encyclopédie Wikipédia, encyclopédie entièrement réalisée par une foule de personnes anonymes ayant la possibilité d'ajouter leur propre information.

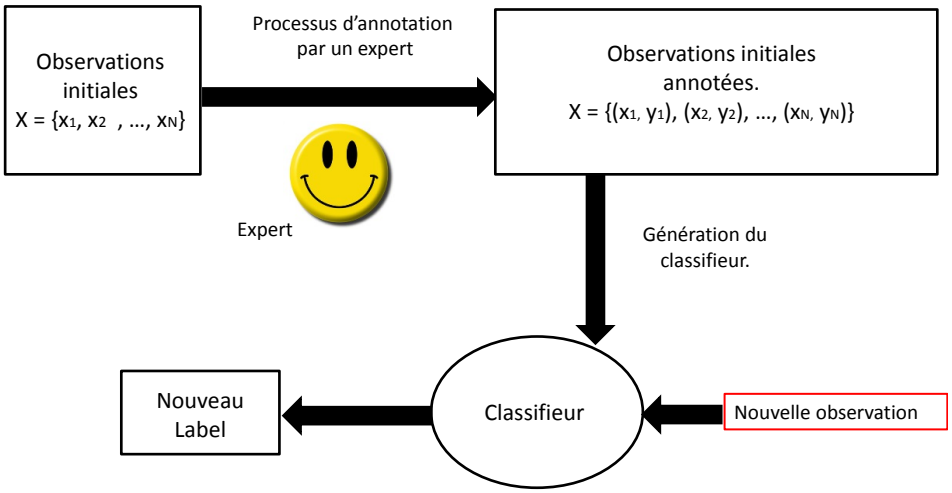


FIGURE 1.2 – Représentation d'un Scénario Classique de Classification Supervisée.

Ainsi, dans un contexte de classification supervisée, le crowdsourcing est une solution très attractive dans le but d'annoter les jeux de données. Pour mieux prendre conscience du potentiel du crowdsourcing, [von Ahn and Dabbish, 2004] ont publié dans leur étude le résultat suivant : une foule de 5000 personnes jouant toute une journée à un jeu en ligne permettrait d'annoter toutes les images indexées par Google (soit 425,000,000 images en 2005) en 31 jours seulement, ce qui est clairement un résultat très prometteur.
Avec le développement récent d'infrastructures telles que Internet, de nombreux services de crowdsourcing ont vu le jour, Amazon Mechanical turk [1] étant le plus connu et le plus utilisé d'entre eux [Snow et al., 2008, Sorokin and Forsyth, 2008, Sheng et al., 2008]. Ces sites permettent alors de faire appel à des centaines, voire des milliers de personnes anonymes sur internet, rendant l'annotation de jeux de données beaucoup plus simple et rapide. Cependant, de nouveaux défis apparaissent suite à leur utilisation, notamment le contrôle de la qualité des données. En effet, si l'utilisation de ces services permet très clairement un gain de temps pour la collecte de données, l'inconvénient majeur de leur utilisation est l'absence totale de moyens pour contrôler la qualité des labels collectés. Cette dernière dépend en effet de plusieurs facteurs : la performance des individus annotant les données, la clarté du problème et des consignes données aux utilisateurs, la qualité des données de départ, etc [Lease, 2011].

Les services de crowdsourcing sont utilisés et étudiés dans de nombreux domaines, tels

1. http ://www.mturk.com.

que en cyber sécurité [Fink et al., 2011] ou en ontologie [Sarasua et al., 2012]. Ici, c'est dans le domaine de l'apprentissage supervisé que nous nous intéressons à l'utilisation de ces services.

1.4 Classification Supervisée et Crowdsourcing

1.4.1 Classification Supervisée Multi-annotateurs

L'acquisition de multiples labels via les sites web de crowdsourcing est très clairement plus facile et rapide que d'obtenir le réel label pour chaque instance du jeu de données entier. Dans ce contexte, si l'on reprend la formulation mathématique adoptée dans la section 1.2.2 pour la classification supervisée, le réel label y_i associé à l'instance x_i est remplacé par de multiples labels $y_i^1, y_i^2, ..., y_i^T$ obtenus par les T annotateurs qui ont annotés le jeu de données via un service de crowdsourcing. Le problème revient alors à un problème de classification en présence de multiples labels. Le Graphique 1.3 présente le scénario de la classification multi-annotateurs.

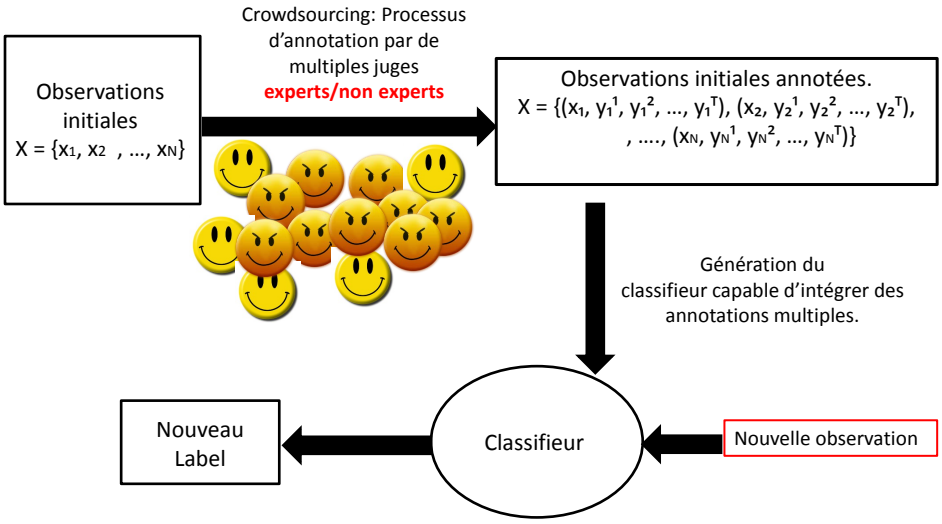


FIGURE 1.3 – Représentation d'un Scénario de Classification en Présence d'Annotateurs Multiples. Dans ce contexte, les annotateurs ne sont pas tous des experts et peuvent être sujet à des erreurs.

La principale problématique dans ce contexte est alors de savoir comment intégrer

ces multiples labels dans la génération du classifieur. Comment transformer les méthodes classiques de classification supervisée afin de prendre en compte les labels des multiples annotateurs ? Une méthode très naturelle et classique pour traiter ce genre de problème pourrait alors être de prendre pour réel label celui qui a été le plus répété parmi tous les annotateurs, puis d'appliquer une méthode de classification classique. Une autre stratégie serait de générer un classifieur pour chaque annotateur, puis d'utiliser le majority voting sur les résultats obtenus afin d'estimer le réel label.
Cependant, toutes ces méthodes supposent que les annotateurs sont experts. Or le principal inconvénient des sites de crowdsourcing est le fait que le processus de labellisation utilisé est très peu contrôlé, les annotateurs ne sont plus des experts, et d'importantes bases de données erronées peuvent être générées. Il est alors important de prendre en compte cette caractéristique lors de la génération du classifieur.

1.4.2 Intégration du Crowdsourcing dans l'Apprentissage

Bien que les web-services facilitent considérablement l'étape d'annotations de jeux de données, ils présentent néanmoins un important inconvénient : tout individu peut participer anonymement à la labellisation des données et aucun contrôle n'est effectué sur ces derniers. Par conséquent, les annotateurs sont souvent non-experts et n'ont généralement pas le même niveau de connaissance dans le domaine concerné. Ils peuvent être incompétents, non expérimentés, et être plus ou moins certains de leurs labels. Dès lors, les labels collectés ne sont pas fiables. Il est alors important de prendre en compte la nature des annotateurs durant la collecte des données et l'incertitude inhérente aux données collectées durant la génération du classifieur. Dans ce contexte, récemment, de nombreuses approches ont été développées pour répondre au problème de classification en présence d'annotateurs multiples. Ces méthodes peuvent être regroupées en plusieurs catégories.
Dans [Whitehill et al., 2009, Raykar et al., 2010], les auteurs génèrent un classifieur intégrant les multiples labels et évaluant la performance de chaque annotateur. L'approche de [Whitehill et al., 2009] permet, en plus, d'estimer la qualité des données.
Dans [Yan et al., 2010], l'auteur suppose que la qualité des labels obtenue dépend non seulement de la performance des annotateurs, mais aussi de la qualité des données de départ. En effet, avec le développement d'Internet, la collecte de jeu de données de grandes dimensions devient de plus en plus facile et rapide. Leur nombre ainsi que leur utilisation évoluent exponentiellement. L'avantage principale de ces jeux de données est d'avoir accès à des centaines voire des milliers de données permettant des analyses plus précises et des résultats plus sûrs. Des conclusions plus pertinentes pourront alors être établies. Néanmoins, l'avantage de ces big data n'est valable que dans la mesure où leur qualité n'est pas détériorée : des jeux de données de grandes dimensions mais de moins bonnes qualités auront pour effet de détériorer les résultats obtenus plutôt que de les améliorer. Il est alors très important de maintenir une bonne qualité de ces jeux de données, ce qui reste aujourd'hui un challenge pour bon nombre d'ingénieurs et de chercheurs.
Enfin, plus récemment, de nouvelles méthodes effectuent une étape de sélection des annotateurs lors de la génération du classifieur. En effet, les spammers (i.e les annotateurs

naïfs) peuvent significativement accroitre le coût d'acquisition des labels et dégrader la qualité du classifieur généré. Par conséquent, un mécanisme les détectant et les éliminant est très clairement désirable dans le but d'améliorer ce dernier, surtout avec l'utilisation récente des web-services. Parmi ces méthodes, on peut citer [Raykar and Yu, 2012, Zhang and Obradovic, 2012].

Bien que toutes ces méthodes aient fait leur preuve dans la littérature statistique, nous considérons ici qu'un point essentiel a été négligé lors de la génération du classifieur : l'incertitude. En effet, le principal inconvénient des méthodes précédemment développées est qu'elles ne donnent pas la possibilité aux annotateurs d'exprimer leur niveau d'incertitude quant au label qu'ils donnent. Par conséquent, tous les labels auront le même poids dans le classifieur, sans faire de distinction entre les labels certains et incertains. De plus, un annotateur conscient de son manque de connaissance sera équivalent à un annotateur inconscient. Or, dans le premier cas, l'annotateur peut être considéré comme un annotateur de confiance lorsqu'il donne un taux de certitude élevé aux labels, contrairement au second. Par conséquent, il nous parait important d'intégrer le degré d'incertitude des juges lors de la génération du classifieur.
Ainsi, l'apport principal de cette thèse est (1) de donner à l'annotateur le moyen d'exprimer son incertitude durant le processus d'étiquetage, (2) de construire des systèmes qui traitent cette incertitude durant l'apprentissage afin d'effectuer une classification performante et robuste, d'évaluer la qualité des instances, et de pouvoir sélectionner les experts parmi les annotateurs.

La section qui suit résume les principales contributions apportées dans cette thèse.

1.5 Contributions

1.5.1 Apport de la Thèse

La première contribution de cette thèse correspond à la possibilité aux annotateurs d'exprimer leur incertitude durant le processus de labellisation. L'expression de cette incertitude est reliée à leur niveau d'ignorance pour le problème posé. Deux cadres sont alors présentés : le cadre de l'incertitude totale et le cadre de l'incertitude partielle. Dans le premier cas, les annotateurs sont complètement ignorants de la réponse à donner et ils attribuent le symbole ' ?' en plus du label. Or ce contexte est une vision simplifiée de la réalité puisque souvent, les annotateurs sont partiellement sûrs des labels qu'ils attribuent. Ainsi, on généralise ce premier cadre à un cadre d'incertitude partielle, où les juges attribuent à chaque label un degré d'incertitude α compris entre 0 et 1. Plus α est proche de 0, plus les annotateurs sont confiants de leur réponse. A l'inverse, plus α est proche de 1, plus les annotateurs sont dans le doute.
Ainsi, dans le cadre de ce travail, chaque juge annote toutes les instances du jeu de données, qu'il soit totalement ou partiellement incertain du label à attribuer. Le symbole ' ?' ou un taux d'incertitude ajouté au label refléteront le degré de confiance accordé à

celui-ci.

La deuxième contribution de la thèse est la construction de trois systèmes permettant d'effectuer une classification performante dans le cadre d'une classification en présence de multiples annotateurs incertains. Le premier modèle proposé, **Ignore**, intègre simultanément la performance des annotateurs ainsi que leur incertitude lors de l'apprentissage supervisé en présence de multiples annotateurs non experts, générant alors un classifieur beaucoup plus stable dans ce contexte.
Le deuxième modèle, **X-Ignore**, se différencie de l'algorithme Ignore par le fait que la phase d'apprentissage évalue, en plus de la performance des annotateurs, la qualité des données. Cette dernière sera alors intégrée dans le modèle lors de la génération du classifieur, en plus de l'incertitude des annotateurs. Enfin, le troisième modèle proposé, **ExpertS**, répond au problème de la sélection des experts durant la phase d'apprentissage. ExpertS filtre les annotateurs les moins performants tout en générant le classifieur uniquement en se basant sur les bons annotateurs.

1.5.2 Méthodologie

Dans un premier temps, on développe les modèles Ignore et X-Ignore dans un contexte de classification binaire, sous la contrainte d'ignorance totale, puis celle de l'ignorance partielle. Ces deux modèles seront ensuite étendus aux cas de classification multiclasses (où plusieurs classes sont disponibles), sous les deux contraintes d'ignorance (totale et partielle). L'extension des modèles Ignore et X-Ignore au cas multiclasses est motivé par le fait que ce contexte de classification est rencontrée dans de nombreuses applications comme par exemple, en traitement du signal et d'images, où un grand nombre de classes peut être disponible [Klautau et al., 2002, Lihong et al., 2009]. Ainsi, chaque modèle Ignore et X-Ignore est composé de 4 versions différentes, suivant le contexte de classification de départ.
Enfin, on s'intéresse au problème de sélection des annotateurs lors de la phase d'apprentissage, avec le développement du système ExpertS.

1.5.3 Modèles Proposés

On résume dans le Tableau 1.1 les différents modèles proposés dans cette thèse. Tous ces modèles ont été testés et validés sur des jeux de données synthétiques de l'UCI Machine Learning Repository [Asuncion and Newman, 2007]. Par ailleurs, nous avons étendu nos expérimentations à une application médicale réelle, dont un descriptif peut être vu à la section suivante.

1.5.4 Application Mélanome

En plus des données synthétiques, les trois modèles proposés ont été testés et validés sur une application médicale réelle, mélanome, dont le but est de prédire, à partir de plusieurs caractéristiques décrivant des images de lésions cutanées, si l'on est en présence

TABLE 1.1 – Résumé des 3 Modèles Proposés

Modèles	Incertitude totale		Incertitude partielle	
	binaire	multiclasses	binaire	multiclasses
Ignore	bIgnore	mIgnore	αbIgnore	αmIgnore
X-Ignore	bX-Ignore	mX-Ignore	αbX-Ignore	αmX-Ignore
ExpertS	binaire			

de mélanomes malins. Le jeu de données a été annoté par 10 dermatologues séniors. Par ailleurs, les labels réels pour chaque image sont disponibles. Cette caractéristique a permis de tester la performance des modèles proposés dans la thèse en comparant le label prédit par nos classifieurs avec la vérité terrain connue.

On résume en détail l'organisation de la thèse ci-dessous.

1.6 Organisation de la Thèse

Cette thèse est organisée comme suit :

- Chapitre 2 : On effectue un bref retour sur les plus importants travaux réalisés dans le domaine de l'apprentissage supervisée en présence de multiples annotateurs, de l'incertitude, de l'estimation de la qualité des données et de la sélection des juges. Trois travaux sont plus largement détaillés ; celui de [Raykar et al., 2010], de [Yan et al., 2010] et [Raykar and Yu, 2012], ces algorithmes représentant les modèles Baselines de la thèse.

- Chapitre 3 : Dans le but de rendre la lecture de cette thèse la plus claire possible au lecteur, on consacre ce troisième chapitre à l'introduction et à l'explication des principales notions techniques utilisées tout au long de notre travail. Le lecteur pourra ainsi se référer à ce chapitre dès lors qu'il en ressentira le besoin.

- Chapitre 4 : Ce chapitre est consacré à la première contribution de notre travail, à savoir le modèle **Ignore**, qui décrit une approche probabiliste bayésienne intégrant l'incertitude des annotateurs lors de la génération du classifieur. Le modèle est dans un premier temps développé dans le cas d'une classification binaire, puis étendu au cas multiclasses par la suite. De plus, on se place pour commencer dans un cas de connaissance ou d'ignorance totale. On verra dans ce même chapitre que le modèle Ignore peut alors très naturellement s'étendre dans le cas d'un degré d'incertitude attribué à chaque label. On compare Ignore à plusieurs modèles précédemment développés, dont le modèle baseline de [Raykar et al., 2010]. Les expérimentations sur des données synthétiques de l'UCI Machine Learning Re-

pository [Asuncion and Newman, 2007] montrent que l'intégration de l'incertitude dans le modèle permet de générer un classifieur beaucoup plus stable et performant face à des juges non experts, en comparaison aux autres modèles n'intégrant pas l'incertitude.

– Chapitre 5 : Ce chapitre est consacré à la deuxième contribution de notre travail, à savoir le modèle **X-Ignore**. X-Ignore décrit une approche probabiliste dans le cas d'une classification supervisée en présence d'annotateurs multiples. L'originalité de cette approche est qu'elle suppose que la performance des annotateurs dépend de leur incertitude ainsi que de la qualité des données de départ. Ainsi, X-Ignore génère un classifieur, estime la performance de chaque annotateur et la qualité de chaque instance présente dans le jeu de données. Comme pour le modèle Ignore, nous décidons pour simplifier de générer tout d'abord le modèle X-Ignore dans le cas d'une classification binaire, puis de l'étendre au cas multiclasses dans un second temps. De plus, on se place pour commencer dans le cas d'incertitude ou de connaissance totale, puis nous étendrons le modèle à un cas plus général où un degré d'incertitude est attribué à chaque label. La performance de X-Ignore est testée et validée sur des données synthétiques de l'UCI, où notre approche est comparée à d'autres systèmes précédemment développés, dont le modèle baseline de [Yan et al., 2010].

– Chapitre 6 : Ce sixième chapitre relève le problème de la sélection des annotateurs lors de la génération du classifieur. En effet, les spammers peuvent significativement accroitre le coût des labels et dégrader la qualité du classifieur généré. Par conséquent, un mécanisme qui détecte les spammers et les élimine est très clairement favorable lors de la génération du modèle. Le modèle **ExpertS** développé dans cette partie combine simultanément la génération d'un classifieur, l'estimation de la performance des annotateurs et l'élimination des spammers. On montre à travers de multiples expérimentations l'efficacité du modèle ExpertS comparée au modèle baseline de [Raykar and Yu, 2012] et à d'autres approches de classification supervisée plus classiques.

– Chapitre 7 : Le dernier chapitre de la thèse a pour objectif de mener une étude appronfondie sur une application médicale réelle *mélanome*, où nos trois contributions Ignore, X-Ignore et ExpertS vont être appliquées afin de comparer les résultats avec une précédente étude menée dans [Wazaefi, 2013]. Les expérimentations effectuées sur ce jeu de données confirment à nouveau la performance et la stabilité de nos modèles en présence de multiples annotateurs non experts. On montre que dans un contexte où les annotateurs sont tous experts, nos modèles sont aussi performants que de précédents modèles Baselines, le réel intêret de nos contributions étant observé dans le cas où de nombreux annotateurs sont non experts : dans ce contexte, nos modèles gagnent très clairement en stabilité.

Chapitre 2

Etat de l'Art

Sommaire

Résumé : Ce chapitre revient sur les principaux travaux réalisés dans le domaine de l'apprentissage supervisée, en commençant par les méthodes de classification supervisée classiques, jusqu'à arriver aux problèmes de classification étudiés de nos jours, à savoir : la classification en présence de multiples annotateurs non experts, la modélisation de l'incertitude dans les modèles, l'estimation et l'intégration de la qualité des données, et enfin les différentes méthodes pour la sélection des annotateurs experts. Trois méthodes, correspondant aux méthodes baselines de cette thèse, sont plus largement détaillées.

2.1 La Classification Supervisée

Le problème du filtrage des spams est un exemple concret d'un problème de classification supervisée, où l'objectif est de distinguer un véritable mail d'un spam. Bien que cette procédure puisse être effectuée visuellement lors de la consultation de la boite mail, elle peut être fastidieuse puisqu'il faudra vérifier chaque mail individuellement, puis regrouper l'ensemble des spams sélectionnés et les stocker dans un autre dossier. Une procédure qui effectuerait cette tâche automatiquement serait alors beaucoup plus appréciée.
Les problèmes de classification supervisée sont présents dans de très nombreux domaines, que ce soit la classification automatique de textes, la catégorisation d'images ou encore le diagnostic de maladies [Gunes and Piccardi, 2006, Blanzieri and Bryl, 2008, Gkanogiannis and Kalamboukis, 2008]. Ainsi, de très nombreuses méthodes ont été développées pour répondre à ce champ d'étude, parmi les plus importantes on retrouve les arbres de décisions, les séparateurs à vastes marges, les K plus proches voisins ou encore la classification naive Bayésienne. On effectue ci-dessous un bref retour sur ces différentes méthodes.

L'algorithme des K plus proches voisins (ou KNN en anglais, K Nearest Neighbors) représente l'algorithme le plus fondamental et le plus simple de la classification supervisée. Dans le but de prédire la classe y d'une nouvelle instance x, l'idée principale de la méthode est de trouver les k plus proches voisins de cette instance, puis d'avoir recours à la classe majoritaire parmi les k plus proches voisins trouvés. Cette méthode a été pour la première fois décrite par [Fix and Hodges, 1951], puis de nombreuses extensions ont alors été publiées, on peut citer [Hastie and Tibshirani, 1996, Klein et al., 2002, Zhang, 2003].

Les algorithmes d'arbres de décision sont apparus pour la première fois aux alentours des années 1960, avec les travaux de [Morgan and Sonquist, 1963]. Ils consistent à prédire la variable catégorielle Y par la construction d'un arbre binaire, où chaque noeud interne de l'arbre représente un critère de discrimination entre les groupes. Ainsi, il suffit de parcourir l'arbre de haut en bas pour prédire la valeur de la classe y d'une nouvelle instance. De très nombreux travaux ont étendu l'algorithme proposé par Morgan et Sonquist, on peut citer par exemple l'algorithme CHAID développé dans [Kass, 1980], ou encore la méthode CART proposée dans [Breiman et al., 1984]. Un exemple d'arbre de décision peut être vu à la Figure 2.1.

Les Séparateurs à Vaste Marge (SVM) ont été introduits par [Vapnik, 1998] à la suite de ses travaux en apprentissage statistique supervisé [Vapnik, 1995]. Cette technique repose sur la recherche d'un hyperplan, qui sépare au mieux les classes entre elles. Elle a tout d'abord été développée dans le cas d'une classification binaire, où l'hyperplan représente une droite, puis étendue au cas multiclasses. Le problème revient alors à trouver une frontière de décision séparant au mieux les différentes catégories. Cette frontière porte le nom d'hyperplan optimal. On parle de maximisation de la marge, où

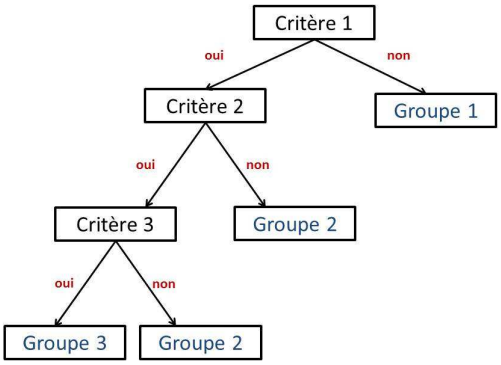


FIGURE 2.1 – Un Exemple d'Arbre de Décision.

la marge se définit comme étant la distance entre le point le plus proche de l'hyperplan et l'hyperplan lui-même.
De très nombreuses techniques de programmation de cette méthode ont vu le jour, telles que l'algorithme publié dans [Platt, 1999], ou plus récemment les algorithmes de Bordes et de Chapelle publiés respectivement dans [Bordes et al., 2005, Chapelle, 2007], ces derniers se focalisant surtout sur l'optimisation du temps d'exécution en traitant des données de grandes dimensions.

Le classifieur bayésien naïf est, comme son nom l'indique, associé au théorème de Bayes, théorème permettant de calculer les probabilités conditionnelles d'un évènement connaissant certains a priori. Ce classifieur repose principalement sur l'hypothèse que les variables sont indépendantes entre elles, ce qui permet alors de simplifier considérablement la détermination des densités de probabilités. Souvent, l'estimation des paramètres de chaque densité de probabilité repose sur la technique du maximum de vraisemblance. Une fois de telles densités estimées, il suffit de calculer les probabilités a posteriori à l'aide de la règle de Bayes pour obtenir le label d'une nouvelle observation. Cette méthode possède plusieurs avantages : elle se montre efficace sur de nombreux jeux de données, comme cela a été montré dans [Hand and Till, 2001]. Elle est simple à programmer, et l'estimation des paramètres y est facile et rapide (même sur des données de grandes dimensions).
Le lecteur pourra se référer au chapitre 3 pour de plus amples informations sur le classifieur bayésien naïf, classifieur qui sera plus tard utilisé dans nos modèles.

Bien que les méthodes habituelles de classification supervisée soient très largement utilisées en apprentissage, leur principal inconvénient est que leur utilisation nécessite la connaissance des labels réels pour l'ensemble des instances du jeu de données de départ. Or ces derniers peuvent parfois être difficiles à obtenir, voir même très coûteux. Le

développement récent d'infrastructures telles que Internet a alors permis l'obtention de multiples labels via des services de crowdsourcing. C'est ce que nous allons étudier dans la section qui suit.

2.2 Crowdsourcing en Classification Supervisée

2.2.1 Les Services de Crowdsourcing

Le terme crowdsourcing a été très récemment introduit par [Howe, 2008], se référant au processus qui vise à externaliser une activité d'une entreprise à un groupe de personnes en réseau. Ainsi, au lieu d'effectuer cette tâche par un nombre de personnes au sein de l'entreprise, cette même tâche pourra être relayée à des centaines voire des milliers d'individus connectés en ligne via des sites spécialisés de crowdsourcing. L'avantage principal de ce procédé est de permettre à l'entreprise d'avoir accès à une large communauté de personnes, chacun d'eux possédant une expérience et une expertise différente des autres, et donnant ainsi la possibilité d'obtenir une très grande quantité de réponses face au problème posé.

De très nombreux services de crowdsourcing se sont alors développés récemment, parmi eux, on retrouve principalement :

- Les services d'annotation d'images : ESP Game [von Ahn and Dabbish, 2004]. Ce jeu consiste à divertir les utilisateurs en leur demandant d'entrer un mot-clé pour chaque image qui apparait. Pour cela, il regroupe les utilisateurs par deux et montre la même image à chaque paire de joueurs. Des points sont alors attribués aux joueurs lorsqu'ils marquent le même mot-clé, les joueurs n'ayant bien entendu pas la possibilité de communiquer entre eux. Ainsi, ESP Game utilise un certain nombre de techniques motivant les utilisateurs à annoter les images sérieusement, et une étude sur la qualité des labels a montré que 85% des labels récoltés sont pertinents dans la description des images [von Ahn and Dabbish, 2004].

- Les encyclopédies en ligne : Wikipédia. Toute personne ayant accès à internet a la possibilité d'éditer ou de modifier des articles dans cette encyclopédie, sans même avoir l'obligation de s'inscrire en créant un compte. Cependant, la pertinence de cette encyclopédie est maintenue par le respect d'une liste de règles très précises, surveillées par un groupe de modérateurs bénévoles. Par exemple, les articles ont l'obligation d'être rédigés d'un point de vue neutre, et avec des informations véridiques. Les adresses IP des personnes malintentionnées sont directement bloquées, retirant ainsi les informations publiées et leur interdisant toute autre action sur le site.

- reCAPTCHA [von Ahn et al., 2008] : système anti-spam permettant la reconnaissance d'images, le déchiffrement de symboles, de mots à travers l'utilisation des

CAPTCHA. Les CAPTCHAs [von Ahn et al., 2004] représentent en effet une famille de tests permettant de différencier l'interaction avec un être humain ou une machine (un ordinateur). La plupart du temps, une suite de lettres ou de chiffres brouillés sont présentés à l'utilisateur, qui doit reconnaitre la séquence CAPTCHA. Ce test permet ainsi de s'assurer que la réponse n'est pas générée par une machine. Les systèmes ReCAPTCHA utilisent les CAPTCHAs dans le but de reconnaitre de nouvelles séquences de mots, d'images, encore inconnues par le système. Ainsi, son principe de fonctionnement est de présenter deux suites de séquences à l'utilisateur, la première étant reconnue par le système, à l'inverse de la seconde. La première séquence permet alors dans un premier temps de passer le test CAPTCHA et d'être ainsi certain que l'on est bien face à un utilisateur. Concernant la deuxième séquence, ReCAPTCHA compare les résultats écrits par tous les utilisateurs. Si un même mot revient à de très nombreuses reprises, alors ReCAPTCHA associe ce mot à la séquence.

– Amazon Mechanical turk : Ce web-service a été introduit en 2005 par Amazon et représente une place de marché qui donne l'opportunité aux entreprises d'accéder à une main d'oeuvre extrêmement large, variée, et à la demande à travers internet. Habituellement, dans le but d'effectuer une tâche précise, l'entreprise aurait recours à un recrutement de salariés temporaires, ce qui peut vite se retrouver être une solution onéreuse en terme de temps et d'argent. Face à ces inconvénients, Amazon Mechanical Turk propose d'accéder à une base de données de milliers de travailleurs. Ce procédé se fait à la demande et à l'avantage d'être à faible coût. Les résultats sont alors intégrés directement via le web-service et deviennent accessibles à l'entreprise. A l'inverse des web-services précédemment décrits, ici les annotateurs sont rémunérés pour le travail demandé. Amazon Mechanical Turk représente aujourd'hui l'un des web-services d'annotations en ligne le plus important et le plus largement utilisé [Akkaya et al., 2010, Laws et al., 2011].

Finalement, de par leur efficacité et leur rapidité, les services d'annotations en ligne sont de plus en plus utilisés dans le but d'annoter des données. Cependant, dans ce contexte, les individus annotant les corpus de données ne sont pas obligatoirement des experts dans le domaine. Ils peuvent avoir des niveaux d'expertises différents, être plus ou moins compétents, ou encore être plus ou moins fiables lors de l'étape d'annotations. Certains peuvent, par exemple, être seulement intéressés par la rémunération du travail. Par conséquent, les labels collectés ont de fortes chances d'être bruités. Ce phénomène a engendré de multiples problématiques : est-ce réellement avantageux d'avoir recours à ces services d'annotations ? Comment faire face aux différences d'expertise entre les annotateurs ? Est-il possible d'identifier les bons annotateurs (experts) des mauvais ?
Récemment, de très nombreux travaux se sont penchés sur ces questions. Dans le traitement automatique du langage naturel, l'étude de Snow et Al., [Snow et al., 2008] a montré que l'utilisation de multiples annotations donne des résultats aussi performants que l'avis d'un seul expert. Ce résultat a été aussi montré dans le domaine de la vi-

sion artificielle [Sorokin and Forsyth, 2008], où les auteurs engendrent des annotations à l'aide du web-service Amazon Mechanical Turk. [Nowak and Rüger, 2010] étudient dans leur travaux la fiabilité de ces services de crowdsourcing. Ainsi, la facilité de nos jours à partager, annoter et organiser des données a engendré de multiples problèmes dans de très nombreux domaines, le domaine de l'apprentissage et de la classification étant directement impliqué.

2.2.2 Crowdsourcing en Classification Supervisée

Dans un contexte plus spécifique d'apprentissage supervisé, le crowdsourcing est une solution très attractive pour annoter les jeux de données. En effet, afin d'appliquer les méthodes classiques d'apprentissage supervisé (cf paragraphe 2.1), il est nécessaire de connaitre le réel label pour toutes les instances du jeu de données. Or dans les applications réelles, l'obtention de ces réels labels peut s'avérer être une étape très coûteuse et fastidieuse. Aujourd'hui, l'utilisation des services de crowdsourcing permet de faciliter cette étape en permettant de collecter des dizaines voir des milliers de labels en très peu de temps. Ainsi, chaque instance est étiquetée par plusieurs annotateurs, ce qui nous place alors dans le contexte de l'apprentissage multi-labels. Dans ce contexte, il est important de prendre en considération le fait qu'ils n'ont pas tous les mêmes compétences et expertises. Certains peuvent être mauvais, d'autres experts, certains annotateurs peuvent aussi être corrélés entre eux. Il est alors essentiel, afin de générer un bon classifieur, de prendre en considération tous ces critères.

Plusieurs approches ont été développées pour répondre au problème de l'apprentissage en présence de multiples annotateurs. Les auteurs de [Dawid and Skene, 1979] et de [Hui and Zhou., 1998] proposent une méthode qui estime les performances des juges, puis qui génère le classifieur. Plus récemment, la communauté s'est focalisée à développer directement des modèles de classification estimant simultanément la performance des annotateurs. Dans cette optique, plusieurs méthodes ont été développées et qui peuvent être regroupées en deux catégories : celles qui utilisent la redondance des labels contre celles qui utilisent des a prioris suivant les similarités entre annotateurs. La première catégorie se base sur l'identification des labels qui améliorent la performance du classifieur. Les méthodes détaillées dans [Smyth et al., 1995, Donmez and Carbonell, 2008, Sheng et al., 2008] font partie de cette catégorie. Ces méthodes sont efficaces lorsque le modèle nous permet de contrôler les labels à prendre en compte lors de la génération du classifieur. A l'opposé, les méthodes qui se basent sur des a prioris estiment les corrélations entre les annotateurs. Cela inclut le travail de [Crammer and Singer, 2003] où les similarités entre les annotateurs et leurs labels sont utilisées afin d'identifier ceux à prendre en compte dans la génération du modèle de classification.

Un autre travail très important et récemment développé est le modèle de Raykar et Al. [Raykar et al., 2010]. Les auteurs présentent une méthode se basant sur l'algorithme EM (Expectation-Maximisation Algorithm) [Dempster et al., 1977] afin d'estimer les véritables labels des instances et d'évaluer la performance des annotateurs. Ce modèle re-

présente un modèle Baseline de notre thèse. On le détaille davantage dans le paragraphe qui suit.

2.2.3 Modèle Baseline de [Raykar et al., 2010]

Le modèle de Raykar et Al. [Raykar et al., 2010] représente un des modèles Baseline de notre thèse. Pour cette raison, on présente brièvement dans cette section les points importants de ce modèle, afin d'introduire certaines notations utilisées dans la suite de ce rapport. De plus, on renvoie le lecteur au chapitre 3 pour un rappel sur certains points mathématiques très souvent utilisés tout au long de cette thèse, à savoir la maximisation de la vraisemblance, l'algorithme EM, ou encore l'algorithme de Newton-Raphson.

Soit N instances x_i et T annotateurs. On note $\mathcal{D} = \left\{\left(x_i, y_i^1, ..., y_i^T\right)\right\}_{i=1}^N$ l'ensemble des labels donnés par les T annotateurs pour l'instance x_i, et z_i son véritable label (inconnu). x_i est un vecteur de dimension d et $z_i \in \{0, 1\}$ dans le cas d'une classification binaire. Soit $p_i = Pr[z_i = 1|x_i]$, la probabilité que l'instance x_i soit dans la classe 1. On définit les matrices $X = \left[x_1^T; ...; x_N^T\right] \in R^{N \times D}$ [1] comme étant la matrice des instances, $Y = \left[y_1^{(1)}, ..., y_1^{(T)}; ...; y_N^{(1)}, ..., y_N^{(T)}\right] \in R^{N \times T}$ matrice des labels des juges et $Z = [z_1, ..., z_N]^T$ le vecteur des réels labels. Les auteurs modélisent le problème avec la distribution conditionnelle jointe $P(X, Y, Z|\Theta)$ où Θ est l'ensemble des paramètres à estimer (définis plus tard), et le but est de maximiser le logarithme de vraisemblance de cette distribution. Pour cela, les auteurs fixent une distribution de Bernoulli pour la variable Y, et une fonction logistique pour Z. Ils se retrouvent finalement à devoir maximiser :

$$ln\ P(X, Y, Z|\Theta) = \sum_{i=1}^{N} z_i\ ln(a_i p_i) + (1 - z_i) ln(1 - p_i) b_i \tag{2.1}$$

avec :

$$a_i = \prod_{t=1}^{T} P[y_i^t | z_i = 1, \alpha^t] = \prod_{t=1}^{T} [\alpha^t]^{y_i^t} [1 - \alpha^t]^{1 - y_i^t}$$

$$b_i = \prod_{t=1}^{T} P[y_i^t | z_i = 0, \beta^t] = \prod_{t=1}^{T} [\beta^t]^{1 - y_i^t} [1 - \beta^t]^{y_i^t}$$

$$p_i = P[z_i = 1|x_i] = \frac{1}{1 + e^{-w^T x}}$$

Finalement, l'ensemble des paramètres à estimer sont $\Theta = \left\{\alpha^1, \beta^1, ..., \alpha^T, \beta^T, w\right\}$ où les deux paramètres $\{\alpha^t, \beta^t\}$ estiment la performance de l'annotateur t en terme de sensitivité et de spécificité, et où le vecteur w représente le poids de chaque descripteur pour la classification.
L'algorithme EM [Dempster et al., 1977] est utilisé pour l'estimation des paramètres

1. $(.)^T$ la matrice transposée

$\left[\alpha^1, \beta^1, ..., \alpha^T, \beta^T\right]$, les labels réels Z étant inconnus. Pour l'estimation de w, les auteurs optent pour l'algorithme de Newton-Raphson [Nocedal and Wright, 2003]. L'algorithme est résumé dans l'algorithme 1.

Algorithm 1 Algorithme d'Apprentissage du Classifieur en Présence de Multiples Annotateurs.

1: Données de départ : X, Y, seuil ϵ, q=0
2: Initialiser : $w^q, \alpha, \beta, \Gamma$
3: Calculer : g, w^{q+1}
4: **while** $||w^{q+1} - w^q||^2 \leq \epsilon$ **do**
5: Algorithme EM : estimation des paramètres $\Theta = \left\{\alpha^1, \beta^1, ..., \alpha^T, \beta^T, w\right\}$.
6: **end while**
7: Retourner la valeur des paramètres $\{\alpha, \beta, w\}$

Les auteurs ont tout d'abord développé cette approche dans le cadre d'une classification binaire, puis l'on étendu au cas multiclasses. Ce modèle a été appliqué avec succès à trois corpus de données réelles : des données de mammographie digitales, des données de cancer du sein, et enfin des données textuelles. Les annotateurs ont été simulés pour les données de mammographie, et 4 radiologistes ont annoté les données de cancer du sein. Concernant les données textuelles, les annotations ont été récoltées via l'utilisation d'Amazon Mechanical Turk. Le lecteur peut se référer à [Raykar et al., 2010] pour de plus amples informations concernant ces jeux de données et les résultats obtenus.

Bien que de nombreuses méthodes ont été adaptées pour répondre au problème de classification en présence de multiples annotateurs, un critère important n'a pas encore été suffisamment pris en considération par ces méthodes : l'incertitude des annotateurs. En effet, bien que les services facilitent considérablement l'étape d'annotations de jeux de données, ils présentent néanmoins un important inconvénient : toute personne peut participer anonymement à la labélisation des données et aucun contrôle n'est effectué sur ces derniers. Par conséquent, leur niveau de compétence est très hétérogène et leur taux d'incertitude a de fortes chances d'être élevé. Ainsi, il est important de prendre en compte ce facteur lors de la génération du classifieur, puisqu'on peut être face à des centaines voire des milliers d'annotateurs incertains. Mais alors une question se pose : de quelle manière peut-on exprimer l'incertitude ? Comment la définir et comment la modéliser ?

2.3 Incertitude, Crowdsourcing et Classification Supervisée

2.3.1 Incertitude dans la Littérature

Le problème de l'incertitude est présent dans de nombreuses applications réelles, telles que en diagnostic médical, en prévisions météorologiques ou encore en apprentissage [Palmer, 2000, Vansteelandt et al., 2006, Haase and Völker, 2005, Liu, 2007]. Mais alors comment définir l'incertitude ? Que représente-t-elle réellement ? Il est en effet important de répondre dans un premier temps à ces questions afin de pouvoir par la suite la modéliser et la traiter. De multiples études se sont penchées sur cette problématique. En économie, [Knight, 1921] étudie la différence entre risque et incertitude, désignant le risque comme une probabilité connue qu'il est possible d'estimer par des données antérieures, contrairement à l'incertitude. Selon lui, l'incertitude est une probabilité inconnue, impossible à déduire, à calculer ou à estimer d'une façon objective. Funtowicz et Ravetz [Funtowicz and Ravetz, 1990] décrivent l'incertitude comme étant une situation où les données sont imprécises ou non fiables, ou encore une situation où il y aurait un manque de connaissance. Cependant, l'incertitude peut aussi avoir lieu dans des situations où la quantité d'informations est très abondante, comme le souligne [van Asselt et al., 2002]. En effet, l'ajout d'un nombre important d'informations peut, par moment, accroitre le taux d'incertitude. Par exemple, de nouvelles connaissances sur un processus complexe peuvent révéler un manque de connaissance autrefois inconnu. [Walker et al., 2003] proposent dans leur article une étude complète sur l'incertitude. Ils définissent une matrice d'incertitude qui distingue trois dimensions de cette dernière :

- Sa localisation : par exemple, serait-elle plutôt due au contexte de l'étude ou plutôt à la qualité des données ?
- Son niveau : il existe un spectre entier représentant plusieurs niveaux d'incertitude, allant de la connaissance totale à l'ignorance totale.
- Sa nature : L'incertitude est-elle due à un manque de connaissance ou à la variabilité naturelle du processus ? Dans le premier cas, on parle d'incertitude épistémique, alors que dans le second cas il s'agit d'incertitude aléatoire.

Ainsi, l'incertitude a fait l'objet de multitudes recherches, d'où le développement de nombreuses méthodes afin de la modéliser et la traiter. On s'intéresse ici principalement à l'incertitude épistémique, puisque cette dernière est associée à l'état de connaissance et est donc prévisible et modélisable. A l'opposé, l'incertitude aléatoire, de par son caractère aléatoire, variable et irréductible, est beaucoup plus difficile à étudier et à modéliser.

2.3.2 Modélisation de l'Incertitude

La modélisation de l'incertitude a reçu depuis plusieurs années, et reçoit toujours une attention tout particulière, et plus spécialement dans le domaine de l'apprentissage. Dans un premier temps, les données incertaines ont été traitées comme étant des données manquantes, où plusieurs méthodes existaient déjà pour leur traitement : exclure tous les individus ayant au moins une donnée manquante, remplacer la valeur manquante par

une valeur plausible, remplacer la valeur manquante par des valeurs provenant d'individus similaires (méthode KNN, K-nearest method). Ces méthodes sont cependant très drastiques, puisqu'elles considèrent une donnée incertaine comme étant totalement absente du corpus ; cela risque donc de supprimer des informations des données recueillies et de détériorer le modèle généré. De plus, les valeurs manquantes sont la plupart du temps dues à de l'aléa, puisqu'elles résultent souvent d'une perte de corpus lors de la collecte d'un très grand nombre de données en un minimum de temps. Or l'incertitude n'est pas due à l'aléa, mais plutôt à un manque de connaissance dans un domaine précis. Ce manque de connaissance peut être plus ou moins fort, et doit donc être d'une certaine manière associé à un degré, où la plus petite (resp. grande) valeur représente une connaissance (resp. ignorance) totale de la réponse. Ainsi, les données manquantes et l'incertitude sont deux notions différentes, et les méthodes utilisées pour les traiter doivent donc être adaptées au problème considéré.

De très nombreux chercheurs se sont intéressés à la modélisation de l'incertitude [Quafafou, 1997, Molenberghs et al., 2001, Imbens and Manski, 2004]. Cette dernière a été très vite associée à la théorie des probabilités, étude mathématique directement liée à l'apprentissage automatique. En effet, la probabilité associée à un évènement reflète le niveau d'incertitude que l'on a face à ce dernier. Cependant, l'inconvénient de cette technique est qu'il faut spécifier une probabilité pour chaque évènement, valeur souvent inconnue dans les situations réelles. Plusieurs méthodes ont alors été développées pour répondre à ce problème, parmi les principales on retrouve :

- **Les approches probabilistes (inférences bayésiennes, réseaux bayésien) :** Cadre idéal lorsque la quantité d'information valable est suffisante pour construire une distribution de probabilité non erronée. Soit un ensemble de N événements possibles $\Omega = \{E_1, E_2, ..., E_N\}$. On associe à chaque événement E_i une mesure de probabilité p_i, telle que $\sum_{i=1}^{N} p_i = 1$. Ainsi, dans le cas où l'avis est donné par un expert, ce dernier renseigne sur son niveau de certitude quant à la réalisation de chaque événement E_i. Dans un cadre bayésien, ce niveau de certitude permet de fixer une distribution a priori, et d'utiliser ainsi les formules de probabilités conditionnelles, en particulier les probabilités bayésiennes avec la règle de Bayes [Bayes, 1763].
  L'un des inconvénients majeurs de cette approche réside dans le fait qu'il faut connaitre parfaitement les probabilités, et plus précisément les probabilités a priori. Or dans la pratique il est souvent très difficile de connaitre ces valeurs, d'où le développement d'autres méthodes telles que la théorie de l'évidence. On invite le lecteur à se reporter au chapitre 3 pour de plus amples détails sur l'approche bayésienne, approche utilisée dans nos modèles par la suite.

- **La théorie de l'évidence :** Théorie de Dempster-Shafer. Cette théorie a tout d'abord été proposée par Dempster, puis complétée par des propositions de Shafer publiées en 1976 [Shafer, 1976]. Elle représente un modèle d'inférence statistique

généralisant l'inférence bayésienne. Elle permet de représenter explicitement l'incertitude liée aux connaissances, et est basée sur la notion de croyance, notion qui se révèle efficace lors de la combinaison de différents points de vue.
Dans sa formulation, soit Ω un référentiel. La connaissance imparfaite est représentée par une fonction $m : 2^{\Omega} \rightarrow [0, 1]$, appelée fonction de croyance. Cette fonction vérifie $\sum_{A_i \subseteq \Omega} m(A_i) = 1$. Les éléments A_i de Ω sont appelés éléments focaux de m. A la différence de la théorie des probabilités, il est possible ici d'attribuer une masse à des sous-ensembles de Ω et pas seulement aux singletons. $m(\Omega) = 0$ correspond à une ignorance complète, alors que la certitude correspond à l'attribution de la totalité de la masse à un singleton de Ω.
La fonction de masse m peut être représentée par deux mesures : la mesure de crédibilité $bel(A_i) : \Omega \rightarrow [0, 1]$ et la mesure de plausibilité $pl(A_i) : \Omega \rightarrow [0, 1]$ respectivement définies par :

$$bel(A_i) = \sum_{B \subseteq A_i} m(B), \ \forall \ A_i \subseteq \Omega$$

$$pl(A_i) = bel(\Omega) - bel(\bar{A}_i), \ \forall \ A_i \subseteq \Omega$$

La mesure de crédibilité de A_i représente ainsi la somme de toutes les masses attribuées aux éléments contenus dans A_i, plus la masse attribuée à A_i lui-même. Pour ce qui est de la plausibilité, elle correspond à la crédibilité maximale possible qui pourrait être attribuée à A_i. Autrement dit, elle représente la somme des masses des éléments ne contredisant pas A_i.

Le principe de la théorie de l'évidence est alors de combiner deux masses indépendantes m_1 et m_2 obtenues de deux experts différents. La combinaison, ou masse jointe, est calculée à l'aide de la règle de Dempster suivante :

$$(m_1 \oplus m_2)(A) = \begin{cases} \frac{1}{(1-K)} \sum_{B \cap C = A_i} m_1(B) m_2(C) & \text{si } A_i \neq \emptyset \\ 0 & \text{si } A_i = \emptyset \end{cases}$$

avec $K = \sum_{B \cap C = \emptyset} m_1(B) m_2(C)$, le degré de conflit.

Ainsi, à travers les fonctions de croyance, la théorie de Dempster-Shafer permet d'évaluer le degré de vérité d'une affirmation et d'étudier sa fiabilité. Cependant, un inconvénient de cette méthode est qu'elle suppose l'indépendance des différentes sources d'informations. Or dans beaucoup de cas, cette hypothèse n'est pas vérifiée puisque les experts peuvent partager leur information, ou encore les classifieurs peuvent être générés sur des méthodes d'apprentissage identiques.

– **Les ensembles flous :** Théorie mathématique développée par [Zadeh, 1965] dans le but de représenter l'imprécision de certaines classes d'objets. En effet, dans la théorie des ensembles classiques, un élément appartient ou n'appartient pas à un sous-ensemble donné. Ainsi, dans ce contexte, seules deux situations sont possibles.

L'originalité de Zadeh a été de tenter de sortir de ce schéma très simplifié de la réalité, et d'introduire la notion d'appartenance pondérée, c'est-a-dire permettre à un objet d'appartenir plus ou moins à un sous-ensemble. Soit l'ensemble de référence Ω et soit E un élément quelconque de Ω. Un sous-ensemble flou A de Ω est défini de la façon suivante :

$$A = \{(E, \mu_A(E)), E \in \Omega\}$$

avec $\mu_A : \Omega \to [0, 1]$. $\mu_A(E)$ représente le degré d'appartenance de E à A. On a alors $\mu_A(E) = 0$ lorsque E n'appartient pas à A, $\mu_A(E) = 1$ lorsque E appartient entièrement à A, et $0 < \mu_A(E) < 1$ si E appartient partiellement à A.
De nos jours, les ensembles flous sont présents dans de très nombreux domaines : en médecine, écologie, recherche scientifique, etc. Pour de plus amples informations, on propose au lecteur de se référer aux ouvrages de références [Dubois and Prade, 1985, Gacôgne, 1997, Bouchon-Meunier and Marsala, 2003].

– **Théorie de Possibilités :** La théorie des possibilités est un autre cadre pour représenter des données incertaines [Zadeh, 1978, Dubois and Prade, 1988]. Elle permet de formaliser des incertitudes de nature non probabilistes sur des évènements. Dans sa formulation, soit X un ensemble fini. On attribue à chaque sous ensemble de X un coefficient compris entre 0 et 1 évaluant la possibilité de cet évènement. On définit alors une mesure de possibilité Π comme étant une fonction sur l'ensemble P(X) des parties de X, qui prend ses valeurs dans [0,1]. Soient A et B deux sous-ensembles de X. On a alors :

$$\Pi(A \cup B) = max(\Pi(A), \Pi(B))$$

$$\Pi(\emptyset) = 0$$

Le nombre $\Pi(A)$ quantifie dans quelle mesure l'évènement $A \subseteq X$ est possible.
On définit de la même façon une mesure de nécessité N, évaluant le degré pour lequel on attend l'occurence d'un évènement. On a alors :

$$N(A) = A - \Pi(\bar{A})$$

Si $N(A) = 1$, alors le sous ensemble A est nécessaire, et est donc certainement vrai. Si, au contraire, $N(A) = 0$, il n'est pas du tout nécessaire que A soit vrai. Si l'on compare la mesure de nécessité avec les probabilités, on peut dire que $N(A) = 1$ entraine $P(A) = 1$, mais $N(A) = 0$ n'entraîne pas forcément que $P(A) = 0$. En effet, même s'il n'est pas nécessaire que A soit vrai, A peut très bien arrivé. A l'inverse, $P(A) = 0$ (évènement A impossible à se réaliser) entraîne forcément que $N(A) = 0$.
Il est aussi possible d'effectuer une comparaison équivalente entre possibilité et probabilité. Par exemple, si $\Pi(A) = 0$ (A est impossible) alors $P(A) = 0$.

Ainsi, on remarque que de nombreuses méthodes ont été développées pour répondre à ce problème d'incertitude. Or ce dernier est aussi très largement présent dans le contexte de classification supervisée en présence de multiples annotateurs, et tout particulièrement avec l'utilisation des services de crowdsourcing. En effet, dans ce contexte, les annotateurs ne sont pas forcément experts, leur manque de connaissance peut être élevé entrainant un niveau d'incertitude significatif. L'incertitude des juges lors de la phase d'annotations peut donc être due à leur faible niveau d'expertise de départ et à leur niveau de connaissances très hétérogène.
Cependant, la faible performance des annotateurs peut aussi dépendre d'un autre critère : la qualité des données qu'on leur présente. En effet, des images floues ou des données incompréhensibles peuvent aussi être la cause du manque de fiabilité des juges.

2.4 Qualité des Données, Crowdsourcing et Classification Supervisée

2.4.1 Introduction à la Qualité des Données

L'étude de la qualité des données n'a cessé de croitre durant ces dernières années, notamment avec le développement récent de certaines infrastructures telles que internet [Pipino et al., 2002, Hubauer et al., 2013]. En effet, de telles infrastructures permettent la collecte, en très peu de temps, de volumes considérables de données, avec cependant un risque qu'elles soient de moins bonne qualité, que ce soit en terme d'erreurs ou de valeurs manquantes.
Or très souvent, la qualité des données est réduite à la notion de précision, c'est-à-dire, à une erreur d'écriture de ces derniers. Par exemple, si l'on prend le prénom "chirine", ce dernier a de très fortes chances d'être écrit avec des erreurs, tels que "Shirine", "Shirin", ou encore "Chitine", toutes ces versions présentant des erreurs par rapport à la version originale. Ainsi, des données sont généralement considérées comme "pauvres en qualité" si leur contenu est mal écrit. Cependant, juger de la qualité des données est une notion bien plus compliquée et profonde que cela. En effet, la régularité, la cohérence, la complémentarité, et la validité dans le temps au moment de l'utilisation sont aussi des dimensions à prendre en considération. Prenons l'exemple du tableau 2.1 : dans ce tableau, les cellules présentant des erreurs sont en gras. Or si l'on se réfère seulement à la notion de précision, seul le livre 3 admet une erreur, où "Du côté de chez Swarn" est écrit au lieu de "Du côté de chez Swann". Cependant, en regardant de plus près, d'autres problèmes sont présents, liés par exemple à un échange d'auteurs entre les livres 1 et 2 ou encore à des renseignements manquants (livre 2). On remarque aussi un problème de cohérence pour le livre 1, où l'année d'édition du livre ne peut pas être antérieure à l'année de naissance de l'auteur. De même, pour le livre 4, l'année de naissance de l'auteur et son année de mort ont été échangées. Ainsi, avec cet exemple, on remarque bien la complexité et l'importance de l'étude de la qualité des données. Cette étude touche très naturellement tous les domaines exposés à l'acquisition et à l'utilisation de données, d'où son importance en statistiques, fouille de données, ou encore en apprentissage au-

TABLE 2.1 – Exemple de Problèmes de Qualité de Données.

Id	Titre du livre	Nom Auteur	Année de Naissance	Année de Mort	Année d'Edition
1	La Nausée	**Camus**	**1905**	1980	**1838**
2	La Chute	**Sartre**	1913	1960	NULL
3	**Du côté de chez Swarn**	Proust	1871	1922	1913
4	Le Procès	Kafka	**1924**	**1883**	1925

tomatique. En effet, l'ensemble de ces spécialités explorent habituellement des jeux de données pouvant être de grandes dimensions, dans le but, par exemple, de trouver des liens entre les différentes variables ou entre les différents individus. Dans cet objectif, il parait naturel que l'acquisition de jeux de données de bonnes qualités soit primordiale. Dans le cas contraire, les résultats finaux seront sujets à de nombreuses interrogations. De ce fait, dans ce travail, il nous a paru important d'intégrer ce critère dans le cadre de la classification supervisée à partir d'annotateurs multiples, notamment avec l'utilisation récente des services de crowdsourcing engendrant la multiplicité des données de grandes dimensions.

2.4.2 Intégration de la Qualité des Données en Apprentissage

Avec le développement d'Internet, la collecte de jeu de données de grandes dimensions devient de plus en plus facile et rapide. Leur nombre ainsi que leur utilisation évoluent exponentiellement. L'avantage principal de ces données est d'avoir accès à des centaines voire des milliers de données permettant des analyses plus précises et des résultats plus sûrs. Des conclusions plus pertinentes pourront alors être établies. Néanmoins, l'avantage de ces big data n'est valable que dans la mesure où leur qualité n'est pas détériorée. En effet, des jeux de données de grandes dimensions mais de moins bonnes qualités auront pour effet de détériorer les résultats obtenues plutôt que de les améliorer. Il est alors très important de maintenir une bonne qualité de ces jeux de données, ce qui reste aujourd'hui un challenge pour bon nombre d'ingénieurs et de chercheurs, dans la mesure où plus les données sont conséquentes, plus il y a de chances à ce qu'elles soient sujet à des erreurs et à des omissions. Ainsi, de très nombreuses études se sont penchées à trouver une mesure pour quantifier la qualité d'un jeu de données, parmi elles on peut citer [Hipp et al., 2001, Pipino et al., 2002].
Or l'utilisation récente du crowdsourcing a donné une nouvelle dimension au problème : la multiplicité des données concerne non seulement le nombre d'exemples dans le jeu de données, mais aussi le nombre de labels récoltés pour chaque exemple. Ainsi, il est encore davantage probable dans ce cas d'obtenir un jeu de données bruité. Plusieurs travaux se sont penchés sur le problème de qualité de données dans un contexte de crowdsourcing. On peut citer parmi eux [Hsueh et al., 2009, Acosta et al., 2013, Sheng et al., 2008]. Des travaux ont aussi été publiés dans le cadre de classification supervisée en présence d'annotateurs multiples, où les auteurs s'intéressent à l'estimation de la qualité des données ou à leur intégration lors de l'apprentissage [Whitehill et al., 2009, Yan et al., 2010].
Dans la section suivante, on détaille l'algorithme développé dans [Yan et al., 2010], correspondant au deuxième algorithme Baseline de notre travail.

2.4.3 Modèle Baseline : [Yan et al., 2010]

Ce modèle, publié dans [Yan et al., 2010], génère le classifieur tout en prenant en considération la performance des annotateurs ainsi que la qualité des données du corpus de départ. Afin de détailler l'algorithme, on reprend les notations adoptées pour le modèle baseline 1 de [Raykar et al., 2010], détaillé dans la section 2.2.3.

Soit $X = \left[x_1^T; ...; x_N^T\right] \in R^{N \times D}$ la matrice des instances du jeu de données, $Y = \left[y_1^{(1)}, ..., y_1^{(T)}; ...; y_N^{(1)}, ..., y_N^{(T)}\right] \in R^{N \times T}$ la matrice des annotations des T juges pour les N instances et $Z = [z_1, ..., z_N]^T$ la matrice des réels labels de chaque instance (ici inconnue). Contrairement au modèle de [Raykar et al., 2010] qui estime la probabilité $P(X, Y, Z)$, ici les auteurs estiment $P(Y, Z|X)$ prenant alors en considération la qualité des données X lors de la génération du modèle. Les auteurs modélisent le problème par la distribution conditionnelle jointe $P(Y, Z|X)$ qui s'écrit :

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

Ils se placent dans un contexte de classification supervisée binaire, et fixent alors un modèle de Bernoulli pour la distribution de y, donnant lieu à l'expression suivante :

$$P(y_i^{(t)}|x_i, z_i) = (1 - \eta^{(t)})^{|y_i^{(t)} - z_i|} {\eta^{(t)}}^{1 - |y_i^{(t)} - z_i|}$$

où le paramètre $\eta^{(t)}$ estime la probabilité que l'annotateur t est raison, c'est-à-dire $y_i = z_i$. Pour s'assurer que le paramètre $\eta^{(t)}$ se situe bien dans l'intervalle]0,1], les auteurs posent :

$$\eta_t(x) = (1 + exp(-w_t^T x_i - \gamma_t))^{-1} \tag{2.2}$$

où le terme γ_t a été ajouté pour s'assurer de la non nullité du paramètre η_t.

Concernant z, les auteurs décident pour des raisons de simplicité d'opter pour la régression logistique :

$$P(z_i = 1|x_i) = (1 + exp(-\alpha^T x_i - \beta))^{-1} \tag{2.3}$$

Les paramètres $\theta = \{\alpha, \beta, \{w_t\}, \{\gamma_t\}\}$ sont alors estimés en maximisant le logarithme du maximum de vraisemblance :

$$argmax_\theta \ \sum_t \sum_i log \sum_{z_i} p(y_i^{(t)}, z_i|x_i; \theta) \tag{2.4}$$

L'équation (2.4) est maximisée à l'aide de l'algorithme EM [Dempster et al., 1977], les labels réels z étant inconnus. L'algorithme de [Yan et al., 2010] est résumé dans l'algorithme 2.

Algorithm 2 Algorithme de Classification en Présence de Multiples Annotateurs avec Prise en Compte de la Qualité des Données.

1: Données de départ : X, Y ; initialiser : $\alpha = 0, \beta = 0$ et le seuil ϵ
2: Initialiser : $\alpha_{new}, \beta_{new}, w_t, \gamma_t$
3: **while** $||\alpha - \alpha_{new}||^2 + (\beta - \beta_{new})^2 \geq \epsilon$ **do**
4: Algorithme EM pour la maximisation de l'expression 2.4 : estimation des paramètres $\theta = \{\alpha, \beta, \{w_t\}, \{\gamma_t\}\}$
5: **end while**
6: **return** $\alpha, \beta, \{w_t\}, \{\gamma_t\}$

Ce modèle a été testé et validé sur des jeux de données synthétiques de l'UCI machine learning repository [Asuncion and Newman, 2007], ainsi que sur deux jeux de données réelles (des données cardiaques ainsi que des données de mammographies).

Le problème d'apprentissage en présence de multiples annotateurs a donné lieu au développement de différentes méthodes ayant fait leur preuve sur de multiples jeux de données. Cependant, un problème n'a pas encore été traité : celui de la sélection des annotateurs. En effet, les annotateurs n'étant pas forcément experts, ils peuvent avoir des niveaux de compétences très hétérogènes. Il serait alors judicieux d'éliminer les moins performants lors de l'apprentissage afin de garantir une meilleure performance du classifieur.

2.5 Sélection des Spammers lors de l'Apprentissage

2.5.1 Introduction au Problème de Sélection des Annotateurs

Plus récemment, le problème d'apprentissage en présence de multiples sources s'est orienté vers la sélection des annotateurs lors de la génération du classifieur. Les juges étant non experts, ils peuvent avoir des niveaux d'expertises différents, certains peuvent être experts alors que d'autres peuvent être peu fiables. Ainsi, il s'avère important de rajouter une étape qui filtre les annotateurs simultanément à l'étape de génération du classifieur, entrainant alors une meilleure utilisation des labels recueillis et la génération d'un classifieur plus performant.
Une méthode très utilisée est d'inclure dans le jeu de données initial des instances dont le label réel est connu, et de pouvoir ainsi évaluer la performance des annotateurs et filtrer les moins compétents. Cette stratégie a été utilisée par le service d'annotations en ligne CrowdFlower. Une autre stratégie serait de faire appel au Majority Voting, où le label le plus répété serait considéré comme le bon [Sheng et al., 2008]. L'inconvénient de cette approche est qu'elle prend en considération la performance d'un annotateur pour un label donné, et non pas pour l'ensemble du corpus. Dans un contexte plus spécifique de classification supervisé en présence de multiples annotateurs, plusieurs travaux très récents ont été publiés utilisant cette méthode dans le but de sélectionner et de donner la priorité aux labels provenant des experts. On peut citer les publications de

[Welinder and Perona, 2010, Yan et al., 2012, Fang et al., 2012].
Un autre travail très récent dans ce domaine est celui de [Raykar and Yu, 2012], proposant d'utiliser une méthode bayésienne, la méthode SpEM, pour éliminer les spammers et estimer les réels labels en se basant uniquement sur les bons annotateurs. Les auteurs introduisent le terme spammers pour désigner les juges annotant les jeux de données aléatoirement, indépendamment de la description des instances données.

2.5.2 Modèle Baseline de [Raykar and Yu, 2012]

Un important travail a était réalisé par [Raykar and Yu, 2012], qui reprennent la méthode développée dans [Raykar et al., 2010] (cf. section.2.2.3) mais en proposant cette fois-ci d'utiliser une méthode bayésienne pour effectuer l'étape de sélection des annotateurs. En effer, leur méthode utilise un a priori ASD (Automatic Detection of Spammer) pour favoriser et éliminer les spammers. Cet a priori admet un paramètre supplémentaire λ, qui sera assigné aux paramètres de performance des annotateurs α et β. Il est alors très important de fixer pour chaque annotateur la bonne valeur du paramètre λ dans le but de ne pas pénaliser les bons annotateurs. Ce paramètre est estimé à l'aide d'une stratégie bayésienne (Maximum de Vraisemblance de Type II, voir [Raykar and Yu, 2012] pour plus de détails). Par conséquent, au lieu de maximiser le logarithme de la vraisemblance, il propose de maximiser le logarithme a posteriori à l'aide de l'estimateur de maximum a posteriori (MAP) :

$$\hat{\Theta}_{max} = argmax \{lnP(X, Y, Z|\Theta) + lnP[\Theta]\}$$

avec $\Theta = \left[\alpha^1, \beta^1, ..., \alpha^T, \beta^T, p\right]$. L'algorithme SpEM est résumé dans l'algorithme.3.

Des expérimentations sur des données simulées et réelles valident la méthode proposée, puisqu'elle est meilleure (ou aussi performante) que des approches précédemment développées en terme de précision, tout en utilisant un nombre d'annotateurs significativement moins élevé.

2.6 Conclusion

Ce chapitre a permis de revenir sur les principales contributions développées dans le domaine de l'apprentissage supervisée jusqu'à aujourd'hui. Nous nous sommes plus particulièrement intéressés aux méthodes développées dans le domaine de la classification supervisée en présence de multiples annotateurs incertains, et nous avons présenté plus précisement les trois modèles baseline de notre thèse, à savoir les approches de [Raykar et al., 2010, Yan et al., 2010, Raykar and Yu, 2012].
Le chapitre qui suit revient sur les principales notions mathématiques abordées tout au long de notre thèse, afin de faciliter la compréhension au lecteur des trois chapitres de contributions proposées par la suite.

Algorithm 3 Algorithme de Détection et Elimination des Spammers dans la Classification en Présence de Multiples Annotateurs.

Données de départ : Annotations $y_i^t, t = 1, ..., T, i = 1, ..., N$.
Initialiser $\lambda^t = 1/N$, for t=1, ..., T.
Initialiser $\mathcal{A} = \{1, ..., T\}$ le jeu des bons annotateurs.
Initialiser $\mu_i = 1/T \sum_{t=1}^{T} y_i^t$ en utilisant le majority voting.
repeat
  repeat
    Recalculer les paramètres p, $\alpha^t, \beta^t, \ \forall t \in \mathcal{A}$, à l'aide de l'algorithme EM.
  until Convergence
  for all t $\in \mathcal{A}$ **do**
    Recalculer λ à l'aide du maximum de vraisemblance type II.
    if $\lambda^t > \delta_1$ (un seuil) **then**
      $\mathcal{A} \leftarrow \mathcal{A} | \{t\}$
    end if
  end for
until changement de l'a posteriori estimé $< \delta_2$
return α, β, p et les spammers détectés dans l'ensemble $\{1, ..., T\} | A$
Estimer le réel label de chaque instance du jeu de données à l'aide du Majority Voting sur l'ensemble des annotateurs sélectionnés.

Chapitre 3

Concept et Outils

Sommaire

Résumé : Ce chapitre est consacré aux principales notions abordées tout au long de cette thèse, afin de permettre au lecteur de s'y référer pour de plus amples explications théoriques. Chaque section de ce chapitre est choisie suivant les principales étapes effectuées pour construire nos classifieurs. Dans un premier temps, on revient sur les différents types de méthodes dédiées à la classification supervisée. Dans une seconde partie, on détaille l'approche probabiliste bayésienne, approche adoptée dans nos contributions pour la modélisation de l'incertitude. Ensuite on effectuera un rappel sur la définition des modèles de mélanges ainsi que des modèles graphiques, afin

d'introduire les distributions jointes dans ce travail. Enfin, on terminera cette présentation par l'étape d'estimation des paramètres, d'où un retour sur l'algorithme EM.

3.1 La Classification Supervisée

La classification supervisée fait toujours intervenir deux sortes de variables : les variables d'entrées $X = \{x_1, x_2, ..., x_N\}$, représentant les N instances du corpus de données initial, et les variables de sorties Y, représentant les N variables à prédire à partir des données X. Le but est alors de modéliser la relation entre ces deux variables X et Y, afin de générer un classifieur pour prédire la valeur y d'une nouvelle instance x. Deux méthodes peuvent alors être envisagées pour répondre à cette problématique : les méthodes discriminatives et les méthodes génératives. Ces deux types d'approches sont détaillés ci-dessous.

3.1.1 Approche Discriminative

L'approche discriminative propose de modéliser directement le problème par la probabilité conditionnelle $P(Y|X)$, l'estimation des paramètres s'effectuant par une étape de minimisation du coût de classification. Nous allons présenter ici les principales approches discriminatives, qui s'avèrent être souvent très efficaces dans les problèmes de classification.

Exemple 1 : La Régression Logistique

On se place dans le cadre d'une classification multiclasses où K classes sont possibles. La régression logistique modélise alors la probabilité conditionnelle $P(Y|X)$ par la distribution suivante :

$$P(Y = k|x_i, \Theta) = \begin{cases} \frac{e^{w_k^T x_i}}{1+\sum_{k=1}^{K} e^{w_k^T x_i}} & \forall k \in \{1, 2, ..., (K-1)\} \\ \frac{1}{1+\sum_{k=1}^{K} e^{w_k^T x_i}} & \text{si } k = K \end{cases} \tag{3.1}$$

où $\Theta = \{w_1, w_2, ..., w_K\}$ sont les paramètres à estimer du modèle. L'estimation de ces paramètres peut alors se faire par la maximisation de la vraisemblance, à l'aide d'algorithmes déjà existants tels que la méthode de Newton-Raphson ou encore la méthode LBFGS (cf. Section 3.4.1 pour de plus amples détails sur ces méthodes).
Un exemple de courbe de régression logistique dans le cas de deux classes peut être vu sur la Figure 3.1.

Exemple 2 : Les SVM

Les Machines à Vecteurs de Support ont été introduits par [Vapnik, 1995], et représentent des méthodes discriminatives non paramétriques, dont la forme générale de la

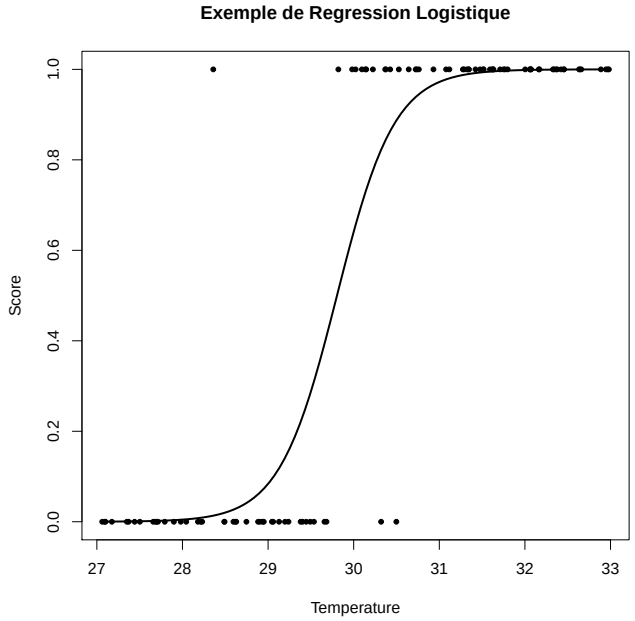


FIGURE 3.1 – Un Exemple de Fonction Logistique : 100 observations simulées, chacune associée à une température et à une classe (0 ou 1). Représentation de la fonction logistique qui prédit la classe d'une nouvelle observation.

distribution est la suivante :

$$f(x) = \sum_i w_i K(x, x_i)$$

w_i sont des coefficients réels des vecteurs supports et K une fonction noyau. L'estimation des coefficients des vecteurs supports peut être résolue par des outils d'optimisation de problèmes convexes classiques. Néanmoins, elle connait certaines limitations, telles que le choix du noyau, qui peut s'avérer être un véritable problème pour de nombreuses applications réelles, ou encore le coût relativement élevé du temps de calcul, comme le souligne [Burges, 1998]. Enfin, les SVM représentent une méthode de classification pour des données binaires seulement ; dans le cas où l'on serait dans un contexte multiclasses, il est nécessaire d'avoir recours à l'algorithme proposé dans [Friedman, 1996], où le problème est transformé en $\frac{k(k-1)}{2}$ classifieurs binaires : chaque classe i étant comparée à chaque classe j. On teste alors toutes les paires de classes.

3.1.2 Approche Générative

Contrairement à l'approche discriminative, l'approche générative cherche dans un premier temps à trouver une structure à la distribution conditionnelle jointe des deux variables X et Y. Elle commence ainsi par modéliser le problème sous la forme $P(X, Y)$, puis en déduit la distribution conditionnelle de Y par l'application de la règle de Bayes

$P(Y|X) = \frac{P(Y,X)}{P(X)}$, avec $P(X)$ la densité des observations initiales. Ce type d'approche comprend par exemple l'Analyse Discriminante Linéaire, ou encore l'Analyse Discriminante Quadratique. Nous renvoyons le lecteur au livre de [Bardos, 2001] pour davantage de détails sur ces deux méthodes. Une autre approche générative est le classifieur de Bayes. Cette méthode ayant été utilisée dans cette thèse, la prochaine section l'étudie plus profondément.

3.2 Approche Probabiliste Bayésienne

Nous avons vu dans le chapitre précédent (cf section 2.3) que la modélisation de l'incertitude peut se faire de plusieurs manières différentes. Nous choisissons dans cette thèse l'approche probabiliste bayésienne, approche qui nous a paru la plus compréhensible et applicable à nos types de données. Ainsi, on effectue ci-dessous un petit retour sur les points importants de l'inférence bayésienne.

3.2.1 Statistique Bayésienne

La statistique bayésienne considère par défaut que les paramètres d'un modèle peuvent prendre n'importe quelle valeur. Il s'avère alors être nécessaire de définir une distribution a priori sur ces paramètres afin de limiter l'aléa. L'ajout d'informations par ces distributions a priori améliore considérablement les performances de la méthode statistique envisagée. Ainsi, autrefois très critiquée pour le caractère subjectif du choix de la loi a priori, l'approche bayésienne est de nos jours un outil reconnu et souvent utilisé en apprentissage. Dans un cadre de données incertaines, elle permet d'apporter les outils nécessaires afin de quantifier et de mettre à jour l'incertitude présente sur le corpus de données. Pour cela, elle se base sur la construction d'une loi a posteriori des variables inconnues, conditionnellement aux données de départ. Cette dernière est généralement calculée en se basant principalement sur la règle de Bayes [Bayes, 1763] : supposons que l'objet inconnu est y connaissant des données x. La densité de probabilité a posteriori est notée $f(y|x)$, et l'utilisation de la règle de Bayes nous donne :

$$P(y|x) = \frac{f(x|y)f(y)}{f(x)}$$

où $f(y)$ représente l'information a priori à introduire. Une des difficultés de cette approche réside alors dans le choix de la distribution a priori. En effet, cette étape est très délicate puisque l'a priori choisi peut avoir une influence non négligeable sur la loi a posteriori. La section qui suit traite de ce problème.

3.2.2 Choix de la Loi a Priori

Le choix des lois a priori est une étape cruciale. Ce choix peut avoir différentes motivations, et les stratégies adoptées sont diverses [Kass and Wasserman, 1996, Gelman, 2006]. Elles peuvent se baser par exemple sur des expériences du passé ou sur une simple intuition. Cependant, le choix de l'a priori reste le point le plus critiquable dans l'analyse

TABLE 3.1 – Exemples de Lois a Priori Conjuguées. Notations : N (loi Normale), G (loi Gamma), B (loi Binomiale) et P (loi de Pareto).

$f(x\|\Theta)$	$\pi(\Theta)$	$\pi(\Theta\|x)$
$\mathcal{N}(\Theta, \tau^2)$	$\mathcal{N}(\mu, \sigma^2)$	$\mathcal{N}(\frac{x}{\tau^2} + \frac{\mu}{\sigma^2}, [\frac{1}{\tau^2} + \frac{1}{\mu^2}]^{-1})$
$\mathcal{G}(n, \Theta)$	$\mathcal{G}(\alpha, \beta)$	$\mathcal{G}(\alpha + n, \beta + x)$
$\mathcal{B}(n, \Theta)$	$\mathcal{B}eta(\alpha, \beta)$	$\mathcal{B}eta(\alpha + x, \beta + 1)$
$\mathcal{P}(\Theta)$	$\mathcal{G}(\alpha, \beta)$	$\mathcal{G}(\alpha + x, \beta + 1)$

bayésienne car il est en pratique très rare que l'information disponible à priori soit suffisamment précise pour conduire à la détermination exacte de la loi a priori. Il s'avère donc très souvent nécessaire de faire un choix arbitraire de cette loi, ce qui peut avoir un impact significatif sur l'inférence qui en découle. Il existe tout de même des lois calibrées en fonction de la distribution des observations de départ, dites lois conjuguées, et des lois à faible contenu informatif, dites lois non-informatives, qui permettent d'évaluer l'influence d'une loi a priori donnée [Box and Tiao, 1973].

Lois a Priori Conjuguées

Dans l'approche bayésienne, l'intégration des informations a priori se fait par le calcul de la distribution dite a posteriori représentant la distribution conditionnelle aux valeurs prises par les données. Cette étape peut s'avérer être très délicate et est facilitée par le recours aux lois a priori conjuguées. En effet, une loi a priori est dite conjuguée si cette dernière et la loi a posteriori ont la même forme.

Definition 1. *Soit π une distribution de probabilité sur les paramètres Θ. π est dite conjuguée si la distribution a posteriori $\pi(\Theta|x)$ appartient à la même famille de distributions que π.*

Le tableau 3.1 montre des exemples de lois a priori conjuguées suivant la distribution f des données x de départ. Dans le tableau, $\pi(\Theta)$ représente la distribution a priori choisie, et $\pi(\Theta|x)$ la distribution a posteriori calculée. On remarque alors que la loi a priori ainsi que la loi a posteriori ont pour chaque exemple la même distribution.

Lois a Priori Non Informatives

Lorsqu'aucune information n'est disponible sur les données, il s'avère difficile de justifier proprement le choix d'une loi a priori sur ces derniers. Dans ce contexte, une solution est de faire appel à une loi a priori non informative, dont le principe est de réduire au maximum l'information apportée par la loi a priori. Elle peut ainsi être définie comme une loi n'apportant pas d'informations et ne donnant pas davantage à une valeur particulière pour le paramètre. Par exemple, soit Θ un ensemble fini de taille q, une loi a priori non informative peut être de la forme :

$$P(\Theta_i) = \frac{1}{q}$$

Il s'agit alors de la loi uniforme. Ici, chaque valeur possible pour Θ se voit attribuer le même poids et les valeurs sont donc équiprobables. Une autre distribution non informative possible est la loi de Jeffreys [Jeffreys, 1946]. Cette méthode utilise l'information de Fisher $I(\Theta)$ représentant une mesure de la quantité d'information sur Θ contenue dans l'observation. Ainsi, plus $I(\Theta)$ est grande, plus l'information apportée par l'observation est grande. Il est alors judicieux de favoriser les valeurs de Θ pour lesquels $I(\Theta)$ est grande, ce qui minimise la loi a priori au profit de l'observation. Ainsi, la régle de Jeffreys consiste à considérer des lois a priori de la forme :

$$\pi(\Theta) \propto \sqrt{detI(\Theta)}$$

où $I(\Theta) = E\left[-\frac{\partial^2}{\partial\Theta^2} log f(x|\Theta)\right]$ dans le cas uni-dimensionnel, avec $E(X)$ l'espérance de X.

3.2.3 Classifieur Bayésien Naïf

La classification naïve bayésienne est principalement basée sur le théorème de Bayes et repose sur une hypothèse forte d'indépendance entre les variables X (d'où le nom de naïf). Pourtant, malgré cette hypothèse, cette méthode se révèle robuste et fiable.
Soit $X = \{x_1, x_2, ..., x_N\}$ l'ensemble des descripteurs, et Y la variable à prédire avec K modalités. L'objectif est alors de maximiser la probabilité a posteriori :

$$\hat{y} = argmax_k \; P[Y = y_k|X]$$

Or d'après la règle de Bayes,

$$P[Y = y_k|X] = \frac{P(Y = y_k) \times P[X|Y = y_k]}{P(X)}$$

Détecter le maximum de cette quantité selon y_k revient à détecter le maximum de $P(Y = y_k) \times P[X|Y = y_k]$, puisque la probabilité $P(X)$ ne dépend pas de Y. Le problème revient donc finalement à maximiser l'expression suivante :

$$\hat{y} = argmax_k \; P(Y = y_k) \times P[X|Y = y_k]$$

L'estimation des paramètres du modèle repose alors très souvent sur le maximum de vraisemblance (cf. Section 3.4.1).

Après cette introduction des approches bayésiennes, il nous faut introduire le modèle probabiliste que l'on souhaite adopter tout au long de notre étude. Or nous introduisons la notion d'incertitude dans nos modèles et on définit alors la notion de deux sous-groupes homogènes qui composent le jeu de données : les données certaines et les données incertaines. Or le processus qui vise à modéliser des sous-groupes homogènes au sein d'un même ensemble de données porte le nom de modèles de mélange. On détaille ainsi dans la partie qui suit le concept des modèles de mélange.

3.3 Modèles de Mélange et Modèles Graphiques

3.3.1 Modèles de Mélange

La notion de modèles de mélange a été introduite par [Pearson, 1894], lors d'une étude portant sur le poids d'un ensemble de crabes : l'histogramme de leur poids présentant une asymétrie, Pearson justifie ce constat par le fait qu'il existe au sein de ces crabes, deux sous-populations de crabes d'origine différente. D'un point de vue plus mathématiques, les modèles de mélange représentent un modèle statistique permettant d'estimer la distribution d'un ensemble de variables aléatoires à l'aide d'une somme de plusieurs distributions, chacune d'elles modélisant chaque sous-groupe du jeu de données.
Soit une variable aléatoire $X = \{x_1, x_2, ..., x_N\}$, et $f(x)$ la densité de probabilité du mélange. Dans le cas d'un modèle discret, $f(x)$ peut alors s'exprimer comme la somme des densités de probabilités $f_k(x)$ de chaque sous-groupe k de X :

$$f(x, \Theta) = \sum_{k=1}^{K} \pi_k \times f_k(x|\Theta_k)$$

où π_k représente la probabilité a priori du sous-groupe k, et où Θ et Θ_k désignent respectivement les paramètres à estimer des distributions f et f_k.
Dans le cas d'une variables aléatoire continue, $f(x)$ s'écrit comme suit :

$$f(x, \Theta) = \int \pi_k \times f_k(x|\Theta_k)dx$$

La probabilité π_k satisfait aux deux règles suivantes : $\pi_k \geq 0$ et $\sum_{k=1}^{K} \pi_k = 1$.
Nous allons voir ci-dessous quelques exemples de modèles de mélange.

1er exemple : Mélange de lois multinomiales. On suppose que les données X suivent un mélange de K lois multinomiales multivariées. La distribution $f(x)$ s'écrit alors :

$$f(x, \Theta) = \sum_{k=1}^{K} \pi_k \times m_k(x, \alpha_k) = \sum_{k=1}^{K} \pi_k \prod_{j,q} (\alpha_k^{jq})^{x^{jq}}$$

où $\Theta = \left\{\pi_1, ..., \pi_K, \alpha_1^{11}, ..., \alpha_K^{JQ}\right\}$, avec :
- π_k : la proportion du sous-groupe k,
- α_k^{jq} : la probabilité que la variable j présente la modalité q dans la classe k.

2ème exemple : Mélange gaussien. Le modèle gaussien est très souvent utilisé dans beaucoup de domaines comme l'apprentissage statistique ou encore la reconnaissance de formes [Goldberger et al., 2003, Guha et al., 1998]. Sa formalisation nécessite juste de remplacer chaque densité f_k par la densité de la loi gaussienne. Soit X une variable de dimension N, on a alors :

$$f(x, \Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(x, \mu_k, \Sigma_k)$$

où

$$\mathcal{N}(x, \mu_k, \Sigma_k) = \frac{1}{(2\pi)^{N/2} \left|det(\Sigma_k)\right|^{1/2}} exp\left(-\frac{1}{2}(x-\mu_k)^t \Sigma_k^{-1}(x-\mu_k)\right)$$

avec μ_k un vecteur de dimension d, Σ_k la matrice de covariance de dimension $d \times d$, et où $|det(\Sigma_k)|$ désigne son déterminant.
Un exemple de mélange gaussien est représenté à la Figure 3.2.

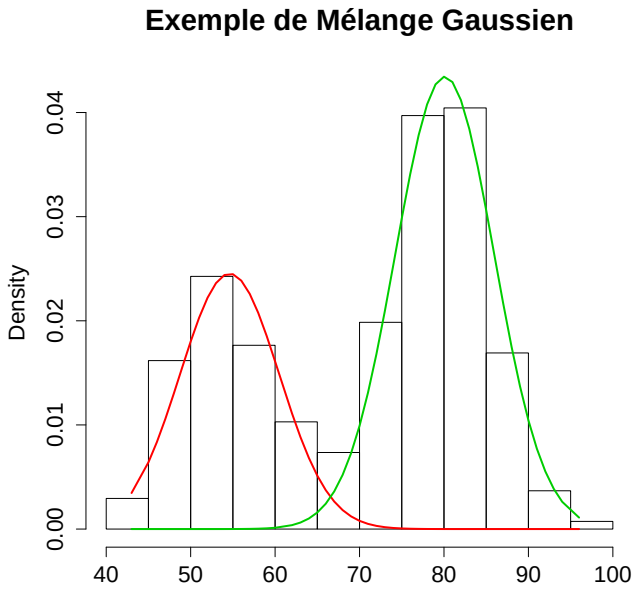


FIGURE 3.2 – Exemple de Modèles de Mélange : Densité d'un mélange de gaussiennes à deux composantes.

Nous allons voir dans la section qui suit un exemple de représentation des modèles de mélange.

3.3.2 Représentation des Modèles de Mélange par les Modèles Graphiques

Les modèles graphiques, encore appelés réseaux de croyances, réseaux probabilistes ou réseaux bayésiens, apportent une interface visuelle grâce à laquelle on peut plus facilement modéliser un problème comportant des variables liées entre elles. Deux éléments essentiels les composent : un graphe illustrant les relations entre les différentes variables du modèle, et un modèle de calcul dérivant du graphe. Les premiers travaux introduisant les modèles graphiques datent des années 1980, avec le travail de [Pearl, 1988]. Dès lors, de multiples possibilités ont été développées, les ouvrages de [Jordan, 1999, Jordan, 2006] en donnant un aperçu. Les principaux avantages des modèles graphiques sont les suivants :

– Ils fournissent un moyen efficace et synthétique pour visualiser la structure d'un modèle,
– Les différentes hypothèses sur le modèle, telles que les relations d'indépendance entre les variables, peuvent être facilement extraites,
– Ils facilitent l'inférence à partir des observations.

Ainsi, les modèles graphiques se situent à la frontière entre la théorie des graphes et la théorie des probabilités. Ils mettent en effet en évidence le lien existant entre un graphe et la notion d'indépendance conditionnelle. On rappelle qu'un graphe est un ensemble de points dont certains sont reliés entre eux par des arêtes. Il existe deux sortes de graphes : les graphes orientés et les graphes non orientés. Dans le premier cas, on distingue le sommet origine de l'arête et son extrémité, contrairement au second cas où les liens sont symétriques (pas de distinction entre les extrémités des liens).
Dans notre situation, on s'intéresse plus particulièrement aux graphes orientés, qui peuvent être définis comme suit :

Definition 2. *Un graphe orienté est noté $G = (S, A)$ avec S l'ensemble des sommets du graphe et A ses arêtes, dont les couples sont des élements de S. Chaque sommet du graphe représente une variable et chaque arête représente une relation de dépendance conditionnelle entre la variable fille et la variable parent.*

La probabilité jointe de toutes les variables d'un graphe orienté est calculée de la façon suivante :

Definition 3. *Soit G un graphe orienté. On suppose que S est composé de l'ensemble des sommets du graphe $\{x_1, x_2, ..., x_L\}$. On note par(x), l'ensemble des parents de $x \in S$. La loi jointe de toutes les variables de S décrite par G s'écrit :*

$$P(x_1, x_2, ..., x_L) = \prod_{l=1}^{L} P(x_l|par(x_l))$$

Prenons l'exemple de la structure du graphe orienté de la Figure 3.3. S est composé de l'ensemble des sommets $\{A, B, C, D\}$. A la lecture du graphe, on peut voir très facilement que A et B n'ont pas de parents, C a comme parents les sommets A et B, et D a comme parent le sommet C. Ainsi, A et B influencent directement sur le sommet C, et ce dernier influence directement sur le sommet D. Il est à remarquer qu'il n'est pas nécessaire de dire que A et B influencent sur D puisque toutes les données relatives à A et à B sont présentes dans C. Enfin, à l'aide de ce graphique, on peut calculer la probabilité jointe $P(A, B, C, D)$ par $P(A)P(B)P(C|A, B)P(D|C)$.
Afin de représenter les modèles graphiques, nous reprenons les conventions décrites dans [Jordan, 1997], cf. graphique 3.4.

Notre but à présent est de représenter les modèles de mélange précédemment décrits par les modèles graphiques. En effet, dans ce contexte, les observations X représentent les données continues observées, Y les classes discrètes non observées et à estimer. Pour

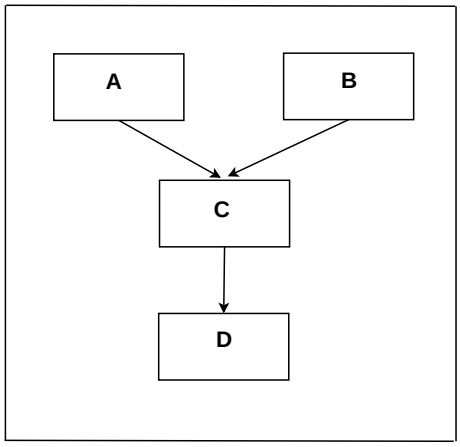


FIGURE 3.3 – Exemple d'un Modèle de Graphe Orienté.

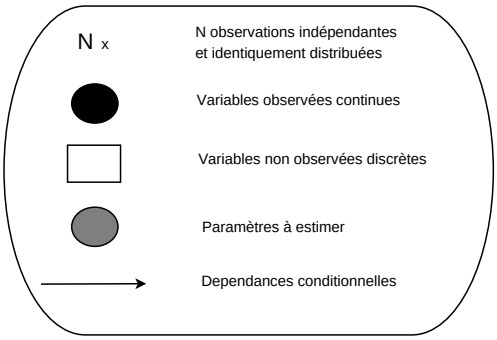


FIGURE 3.4 – Conventions de Représentation des Modèles Graphiques.

chaque classe Y, une probabilité a priori π est envisagée, et les données X suivent une distribution de paramètres Θ à estimer, cf. Figure 3.5.

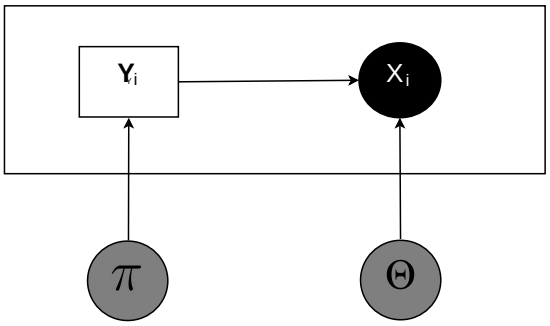


FIGURE 3.5 – Modèle Graphique d'un Modèle de Mélange.

Le but est alors d'estimer le paramètre Θ du modèle. Différentes techniques peuvent alors être utilisées, mais la présence de paramètres inconnus dans notre contexte nous amène à choisir l'algorithme EM (Expectation Maximisation Algorithm). La section qui suit effectue un rappel sur la théorie de cet algorithme.

3.4 Algorithme EM

L'algorithme EM (*Expectation Maximisation algorithm*) [Dempster et al., 1977] représente une approche fondamentale en apprentissage statistique puisqu'il s'agit de la solution la plus couramment utilisée pour des problèmes d'apprentissage en présence de variables non observées. En effet, dans la pratique, les données obtenues des applications réelles présentent très souvent des données manquantes. Or la plupart des méthodes de classification automatique font l'hypothèse que toutes les données sont présentes. Ainsi,

très souvent, les instances présentant des données manquantes sont ignorées ou prétraitées, ce qui est loin d'être une solution optimale puisque cela peut introduire à un biais dans l'analyse. Ainsi, de par sa simplicité et sa capacité à résoudre nombre de problèmes d'estimations différents, l'algorithme EM est devenu incontournable en apprentissage statistique. Le plus souvent, l'algorithme est utilisé lors de la phase d'estimation des paramètres du modèle, où la maximisation de la vraisemblance en présence de données manquantes est difficile. Nous effectuons ainsi pour commencer un bref retour sur la notion de vraisemblance et sur sa maximisation.

3.4.1 Maximum de Vraisemblance

Développé par le statisticien Ronald Aylmer Ficher en 1922 [Fisher, 1922], l'estimation du maximum de vraisemblance est couramment utilisée pour inférer les paramètres d'une distribution de probabilité d'un échantillon donné. Dans cette section, nous revenons sur le principe général de la méthode, et nous détaillons certains problèmes numériques liés à la résolution de sa maximisation, techniques que nous utiliserons plus tard dans nos propositions.

Estimation par Maximum de Vraisemblance

Soit X une variable aléatoire réelle, discrète ou continue, de paramètre Θ inconnu à estimer. On définit alors une fonction f telle que :

$$f(x;\Theta) = \begin{cases} f_\Theta(x) & \text{si X est une variable aléatoire continue,} \\ P_\Theta(X = x) & \text{si X est une variable aléatoire discrète.} \end{cases} \tag{3.2}$$

On appelle $f_\Theta(x)$ la densité de X et $P_\Theta(X = x)$ une probabilité discrète. On observe une réalisation de X de taille n, soit $x = \{x_1, x_2, ..., x_n\}$. On suppose que ces observations sont indépendamment et identiquement distribuées (iid).

Definition 4. *On appelle fonction de vraisemblance, la fonction de x et de Θ telle que :*

$$V(x_1, x_2, ..., x_n; \Theta) = \prod_{i=1}^{n} f(x_i; \Theta)$$

Une estimation $\hat{\Theta}$ des paramètres Θ est alors obtenue en maximisant la vraisemblance.

Definition 5. *On appelle estimateur de maximum de vraisemblance, la valeur $\hat{\Theta}$ solution du problème de maximisation :*

$$\hat{\Theta} = max_\Theta \; V(x_1, x_2, ..., x_n; \Theta)$$

Une transformation croissante ne changeant pas le maximum d'une fonction, elle est très souvent considérée pour simplifier le calcul le logarithme népérien de la vraisemblance, ce qui nous donne :

$$\hat{\Theta} = max_\Theta \; Log(V) = max_\Theta \; (LV(x_1, x_2, ..., x_n; \Theta))$$

Finalement, le problème de maximisation de la vraisemblance revient à un problème de recherche d'optimums. De nombreuses méthodes ont été développées dans ce contexte, et c'est ce que nous allons présenter dans la section qui suit.

3.4.2 Quelques Méthodes d'Optimisation

Nous présentons ici un ensemble de méthodes permettant de résoudre la recherche d'optimums de fonctions. En effet, de nombreuses applications ont recours à la minimisation ou à la maximisation d'une fonction. Une méthode simple pour résoudre ce problème serait d'utiliser le fait que si LV est dérivable et si LV admet un maximum global en une valeur $\Theta = \hat{\Theta}$, alors la dérivée première s'annule en $\Theta = \hat{\Theta}$ et la dérivée seconde est négative. Réciproquement, si la dérivée première s'annule en $\Theta = \hat{\Theta}$ et que la dérivée seconde est négative en ce point, alors $\Theta = \hat{\Theta}$ est un maximum local (et non global) de $LV(x_1, x_2, ..., x_n; \Theta)$. Reste alors à vérifier qu'il s'agit bien d'un maximum global. Dans la pratique, cette technique a cependant souvent l'inconvénient d'être très complexe à réaliser : en effet, LV n'est pas tout le temps dérivable, et peut avoir une forme analytique complexe entrainant des problèmes d'existence et d'unicités, mais aussi des problèmes de calculs numériques non aisés à résoudre. Des méthodes ont alors été développées pour répondre à ce problème. Nous détaillons dans ce paragraphe quelques-unes de ces méthodes, et plus particulièrement celles de Newton-Raphson et LBFGS, méthodes que nous avons utilisés dans notre travail [Nocedal and Wright, 2003]. Néanmoins, on invite le lecteur à se référer au livre de [Baranger, 1977] pour davantage approfondir ce sujet.

Soit f la fonction à minimiser, c'est-à-dire dans notre cas le log de la vraisemblance. Une technique générale est de rechercher tout d'abord les extremums locaux de f puis de déterminer l'extremum global. Dans la suite, on note $f^{'}$ la dérivée de f. On rappelle que si x_0 est l'extremum de f, alors $f^{'}(x_0) = 0$. C'est cette propriété que la plupart des méthodes utilisent pour résoudre le problème d'optimisation.

Méthode de balayage :

On fixe un pas de déplacement sur un intervalle donné a priori, de telle manière que la division soit équidistante, et on calcule successivement les valeurs de la fonction à minimiser en ces intervalles, en incrémentant du pas de balayage. Cette méthode est très basique et sommaire, et est surtout réservée au cas unidimensionnel. Les avantages de cette méthode est qu'elle permet de distinguer entre les minimums locaux et le minimum global. Par contre on peut remarquer l'importance du pas choisi, puisque le résultat final dépend de ce dernier.

Méthode de dichotomie (ou de la bissection) :

Soit deux nombres a et b, et on considère $f^{'}$ continue sur l'intervalle [a,b]. On suppose que $f^{'}(a)$ et $f^{'}(b)$ sont de signes contraires. Ainsi, d'après le théorème des valeurs intermédiaires, $f^{'}$ a au moins un zéro dans l'intervalle [a,b]. La méthode de dichotomie

consiste alors à diviser l'intervalle en deux en calculant $c = (a+b)/2$. A partir de là, soit $f^{'}(a)$ et $f^{'}(c)$ sont de signes contraires, ou $f^{'}(c)$ et $f^{'}(b)$. L'algorithme de dichotomie est alors appliqué au sous-intervalle dans lequel le changement de signe a eu lieu, entraînant un algorithme itératif.

Méthode de Newton :

La méthode de Newton cherche à construire une bonne approximation de la racine de la fonction $f^{'}$ en se basant sur le développement de Taylor. Autrement dit, partant d'un point de départ x_0 que l'on choisit de préférence proche du zéro à trouver, on considère la fonction a peu près égale à sa tangente en ce point :

$$0 = f^{'}(x) \approx f^{'}(x_0) + f^{''}(x_0)(x - x_0)$$

où $f^{'}$ désigne la dérivée de f et $f^{''}$ sa dérivée seconde. En résolvant cette équation, on obtient un point x_1, qui sera plus proche de la valeur de la racine que le point x_0 précédent. Ainsi, plus généralement, on part d'un point x_0 fixé au départ, et on construit par récurrence la suite :

$$x_{k+1} = x_k - \frac{f^{'}(x_k)}{f^{''}(x_k)}$$

Cette suite converge vers la racine de $f^{'}$ et donc vers le minimum de f. Le lecteur peut se référer aux articles et ouvrages suivant pour de plus amples informations [Whiteside, 1974, Nocedal and Wright, 2006, Bonnans et al., 2006].

On remarque que les méthodes citées ci-dessus sont des méthodes de minimisation adaptées à des fonctions de dimension 1. On détaille ci-dessous certaines méthodes dans le cas multidimensionnel.

Méthode de Newton-Raphson :

Il s'agit de l'application directe de la méthode de Newton, dans le cas multidimensionnel. De façon formel, on note G(f) le vecteur gradient de la fonction f à minimiser, et H(f) sa matrice hessienne des dérivées partielles secondes. La méthode consiste à réaliser une suite d'itérations de la forme :

$$x_{k+1} = x_k - (H(f(x_k)))^{-1} G(f(x_k))$$

Cette méthode est efficace pour la minimisation d'une fonction convexe. Sa mise en oeuvre est simple, sauf si sa matrice hessienne est singulière. D'autres part, elle nécessite énormément de calculs et beaucoup de mémoire, puisqu'à chaque itération, le gradient et l'hessien sont à calculer et à stocker.

Méthodes de Quasi-Newton :

En pratique, souvent, le hessien d'une fonction est très difficile à évaluer dans le cas ou f n'est pas analytique. Le gradient est plus facilement accessible. On souhaite donc ne pas avoir à calculer la valeur exacte du hessien, mais simplement à évaluer une approximation. Ces méthodes quasi-Newtoniennes sont donc semblables à la méthode de Newton, à la différence qu'elles ne calculent pas explicitement la matrice hessienne, mais une approximation qui peut être modifiée et corrigée à chaque itération. Il existe plusieurs méthodes Quasi-Newtonienne, suivant l'approximation choisie. On pose, à l'itération n, $s_n = x_{n+1} - x_n$ et $y_n = G(f_{n+1}) - G(f_n)$. Dans la méthode BFGS (Broyden, Fletcher, Golfarb, Schanno), l'approximation de l'hessien $\hat{H}_{n+1}$ est donnée par :

$$\hat{H}_{n+1} = \hat{H}_n + \frac{y_n y_n^T}{y_n^T s_n} - \frac{\hat{H}_n s_n s_n^T \hat{H}_n}{s_n^T \hat{H}_n s_n}$$

Cette méthode a été améliorée par la méthode LBFGS (Limited-memory BFGS) , par le fait qu'elle ne requiert le stockage que d'un nombre beaucoup plus petit de variables. Elle utilise pour cela quelques vecteurs qui représentent l'approximation implicitement. Ainsi, LBFGS est souvent utilisé dans des problèmes d'optimisation à grande dimension [Nocedal and Wright, 2003].
Une autre méthode quasi-Newton est la méthode DFP (Davidson, Fletcher, Powell) qui estime quant à elle l'hessien par l'expression suivante :

$$\hat{H}_{n+1} = \hat{H}_n + \frac{s_n s_n^T}{s_n^T y_n} - \frac{\hat{H}_n y_n y_n^T \hat{H}_n}{y_n^T \hat{H}_n y_n}$$

L'ensemble des méthodes de minimisation citées ci-dessus ne prennent pas en compte la possibilité de données manquantes. Ainsi, elles ne sont pas utilisables dans ce cas-là, et l'algorithme EM devient alors une alternative pour répondre au problème.

3.4.3 Algorithme EM Classique

L'algorithme EM [Dempster et al., 1977] est un algorithme décomposé principalement en deux étapes (d'où son nom) :

- Etape E : Etape d'évaluation de l'espérance. Lors de cette étape, l'information actuellement disponible (c'est-à-dire les données observées mais aussi l'estimation ponctuelle des paramètres) est utilisée pour estimer les valeurs manquantes.
- Etape M : Etape de maximisation. Lors de cette étape, les paramètres du modèle sont estimés en maximisant la vraisemblance à l'aide des distributions de probabilités obtenues à l'étape précédente.

Les étapes E et M sont appliquées de façon itérative jusqu'à ce qu'il y ait convergence de l'algorithme. D'un point de vue plus mathématique, supposons les deux ensembles de variables X et Y, où $X \in \mathcal{X}$ représente le groupe de variables observées et $Y \in \mathcal{Y}$

représente le groupe de variables manquantes. Soit Θ les paramètres à estimer du modèle définis sur $\mathcal{X} \times \mathcal{Y}$. La densité marginale de X est alors donnée par :

$$p(x|\Theta) = \sum_{y \in \mathcal{Y}} p(y|\Theta) p(x|y, \Theta) \tag{3.3}$$

En supposant que les observations $\mathrm{X} = \{x_1, x_2, ..., x_N\}$ sont iid (indépendantes entre elles et identiquement distribuées), la fonction à maximiser au sens du maximum de vraisemblance pour obtenir une estimation $\hat{\Theta}$ des paramètres est :

$$V(\Theta, X) = \prod_{i=1}^{N} P(x_i; \Theta) = \prod_{i=1}^{N} \sum_{y \in \mathcal{Y}} P(y; \Theta) P(x_i|y; \Theta) \tag{3.4}$$

La maximisation de cette expression passe dans un premier temps par la décomposition de la loi jointe entre variables observées et variables latentes sous la forme d'un produit entre une probabilité conditionnelle et une probabilité a priori :

$$P(x, y|\Theta) = P(y|x, \Theta) P(x|\Theta) \tag{3.5}$$

Le passage au logarithme donne :

$$log\left(P(x, y|\Theta)\right) = log\left(P(y|x, \Theta)\right) + log\left(P(x|\Theta)\right) \tag{3.6}$$

Ce qui amène directement à l'expression suivante pour le log de la vraisemblance :

$$log\left(P(x|\Theta)\right) = log\left(P(y|x, \Theta)\right) - log\left(P(x, y|\Theta)\right) \tag{3.7}$$

A l'itération q de l'algorithme, connaissant les données observées et en supposant connaître la valeur courante des paramètres, l'espérance du log de la vraisemblance par rapport à la loi conditionnelle des variables latentes est :

$$\begin{aligned} E\left[log\left(P(x; \Theta)\right) | X = x; \Theta = \Theta^q\right] = {} & E\left[log\left(P(x, y; \Theta)\right) | X = x, \Theta = \Theta^q\right] \\ & - E\left[log\left(P(y|x; \Theta)\right) | X = x, \Theta = \Theta^q\right] \end{aligned}$$

Or $log\left(P(x; \Theta)\right)$ ne dépend pas de Y. Donc on a :

$$E\left[log\left(P(x|\Theta)\right) | X = x, \Theta = \Theta^q\right] = log\left(P(x|\Theta)\right) = V(\Theta, x) \tag{3.8}$$

On pose :

$$\begin{aligned} Q(\Theta, \Theta^q) &= E\left[log\left(P(x, y|\Theta)\right) | X = x, \Theta = \Theta^q\right] \\ &= \sum_{y \in \mathcal{Y}} P(y|x, \Theta^q) log(p(x, y|\Theta)) \end{aligned}$$

$$\begin{aligned} H(\Theta, \Theta^q) &= E\left[log\left(P(y|x, \Theta)\right) | X = x, \Theta = \Theta^q\right] \\ &= \sum_{y \in \mathcal{Y}} P(y|x, \Theta^q) log(p(y|x, \Theta)) \end{aligned}$$

H se dégrade naturellement au cours des itérations de l'algorithme (cf. Annexe A), et il suffit alors de maximiser la quantité Q pour augmenter la vraisemblance V. Ainsi, à chaque itération, l'algorithme EM augmente la vraisemblance des paramètres à estimer. L'algorithme 4 résume les différentes étapes de l'approche EM :

Algorithm 4 Pseudo-code de l'Algorithme EM

1: Données en entrées : Données initiales X connues.
2: Paramètres à initialiser : Θ^0, Θ^1, étape q=0..
3: **while** test d'arrêt **do**
4: q=q+1
5: Etape E : Calculer $Q(\Theta, \Theta^q)$.
6: Etape M : Calcul de Θ^{q+1} en maximisant l'équation obtenue à l'étape E :

$$\Theta^{q+1} = argmax_{\Theta}\ Q(\Theta, \Theta^q)$$

7: **end while**
8: Retourner les paramètres finaux de Θ estimés.

Remarques :

1. La convergence de l'algorithme EM vers un maximum local a été démontrée dans plusieurs études [Dempster et al., 1977, Wu, 1983, Xu and Jordan, 1995]. Cependant, l'initialisation des paramètres joue un rôle très important dans la convergence. Ainsi, souvent, il est nécessaire de répéter l'algorithme avec différentes valeurs d'initialisation et de choisir celle avec les meilleurs résultats finaux, en espérant que la convergence qui s'en suit corresponde bien au maximum global de la vraisemblance. Il existe par ailleurs d'autres stratégies qui ont été mis au point suivant le problème traité [Biernacki et al., 2003].
2. L'algorithme EM étant itératif, il est nécessaire de fixer un test d'arrêt. Différentes solutions sont alors possibles : regarder l'évolution de la fonction de vraisemblance, l'évolution des probabilités a posteriori ou encore l'évolution des valeurs des paramètres.

3.4.4 Algorithme EM et ses Extensions

L'algorithme EM a été très largement étudié et adapté suivant les problèmes rencontrés. Ainsi, de nombreuses extensions ont été proposées. Nous citons ici quelques-unes d'entre elles, le lecteur pourra se référer à l'ouvrage de [McLachlan and Krishnan, 2008] pour un plus grand tour d'horizon de ses extensions et de ses applications.

Réduction du Temps de Convergence

Un des premiers problèmes rencontrés avec l'algorithme EM classique est le temps nécessaire à sa convergence. En effet, avec l'évolution des infrastructures telles que internet, il s'avère être de nos jours très courant de devoir travailler avec des données de

très grandes tailles. Beaucoup de travaux se sont alors intéressés à des méthodes dans le but de minimiser le temps de convergence de l'algorithme EM, sans pour autant toucher à sa simplicité, son efficacité et à sa stabilité.
Dans les situations où l'étape M s'avère compliquée, comme par exemple le cas où le modèle compte de nombreux paramètres à estimer, il peut être plus simple de diviser le problème en plusieurs sous-problèmes : le fait de connaitre la valeur de certains paramètres pourrait en effet aider à trouver les valeurs des paramètres restants. Cette technique a été utilisée par [Meng and Rubin, 1993], où ils présentent l'algorithme ECM, qui remplace l'étape de maximisation par plusieurs étapes de maximisation conditionnelle. Par conséquent, dans de nombreuses applications, ECM dépasse en terme de rapidité et de stabilité l'algorithme EM classique. Parmi les autres approches considérées, l'étape M peut aussi se voir être remplacée par la méthode quasi-Newton [Meilijson, 1989], ou encore la méthode du gradient conjugué [Jamshidian and Jennrich, 1993].
Dans les situations où l'étape M est relativement simple, le temps de convergence est surtout optimisé durant l'étape E, où en temps normal cette étape calcule chaque instance individuellement. [Neal and Hinton, 1998] ont proposé l'algorithme EM en version incrémenté (IEM), où les n observations de départ sont divisées en B blocs. L'étape E est alors implémentée pour un block à la fois, avant de procéder à l'étape M. Ainsi, IEM est composé de B étapes partielles de E et de B étapes M. IEM a ainsi plus de rapidité à intégrer de nouvelles observations, plutôt que de lire toutes les données en une seule fois avant de procéder à l'étape M. Il faut cependant être vigilant quant à la valeur B de blocs à créer puisque au-delà d'un certain seuil, le temps de convergence va recommencer à augmenter [Ng and McLachlan, 2003]. Il est alors important de choisir la valeur de B qui est appropriée pour conserver la performance de l'algorithme IEM.
Une autre solution proposée par [Neal and Hinton, 1998] pour accélérer le temps de convergence est l'algorithme SPEM (Sparse EM Algorithm), où les auteurs montrent qu'en optant pour un modèle de mélange pour le jeu de données par maximum de vraisemblance via l'algorithme EM, les estimations a posteriori de certains paramètres seront souvent proche de zéro. Avec l'algorithme SPEM, ces probabilités a posteriori sont maintenues fixes, et seulement les probabilités a posteriori des autres paramètres sont estimées de nouveau.

Algorithme EM dans le Cas Bayésien

L'algorithme EM peut s'appliquer dans le cadre bayésien très naturellement dans le but de maximiser la distribution a posteriori $\pi(\Theta|y)$. A l'aide du théorème de Bayes et en passant au logarithme, l'estimateur de maximum a posteriori revient à maximiser :

$$log\pi(\Theta|y) = logL(\Theta) + log\ \pi(\Theta)$$

Par application de l'algorithme EM, on obtient :
– Etape E : à l'itération (k+1) calculer :

$$E_{\Theta^k}\left\{log\pi(\Theta|x)|y\right\} = Q(\Theta, \Theta^k) + log\ \pi(\Theta)$$

L'étape E ne change pas dans le cadre bayésien, on se retrouve à devoir calculer $Q(\Theta, \Theta^k)$.

– Etape M : Maximiser $Q(\Theta, \Theta^k) + log\ \pi(\Theta)$. L'étape M diffère de l'algorithme EM classique, puisqu'il y a un terme supplémentaire venant de l'a priori. Cependant, la présence de ce terme implique une fonction plus concave et permet d'accroitre la vitesse de convergence de l'algorithme.

Algorithme GEM

L'étape de maximisation peut s'avérer être difficile à réaliser, et il peut ne pas exister de solution analytique. Dans ce cas, cette étape peut être partiellement effectuée dans la mesure où la croissance de Q suffit à garantir la croissance de la vraisemblance. Cette variante est nommée algorithme GEM (Generalized EM algorithm) et a été proposée en même temps qu'EM par [Dempster et al., 1977]. A l'étape M, Θ^{q+1} est choisi de tel manière que :

$$Q(\Theta^{q+1}; \Theta^q) \ \geq \ Q(\Theta^q; \Theta^q)$$

Ainsi, dans cette version de l'algorithme, Θ^{q+1} ne doit qu'accroitre la fonction $Q(\Theta; \Theta^q)$ avec $\Theta = \Theta^q$, au lieu de l'accroitre sur toutes les valeurs possibles de Θ.

De nombreuses autres extensions ont été proposées pour l'algorithme EM, par exemple dans le cas particulier de modèle de mélange. C'est ce que nous allons voir dans la section suivante.

3.4.5 Algorithme EM dans le Cadre de Modèles de Mélange

L'algorithme EM s'applique très facilement aux modèles de mélange. En effet, l'adaptation de cet algorithme à ce contexte ne change pas le calcul de l'étape E, où les probabilités a posteriori sont calculées de la façon suivante :

$$\mu_{ik}^q = P(y_i = k | x_i, \Theta^q) = \frac{\pi_k^q f(x_i; \Theta_k^q)}{\sum_{k'=1}^K \pi_{k'}^q f(x_i; \Theta_{k'}^q)} \tag{3.9}$$

Seule la fonction Q à maximiser à l'étape M change dans ce cadre, où elle prend la forme suivante :

$$Q(\Theta, \Theta^q) = \sum_{i=1}^N \sum_{k=1}^K \mu_{ik}^q log(\pi_k f(x_i; \Theta_k)) \tag{3.10}$$

Ainsi, la seconde étape de l'algorithme EM dépend des choix des densités conditionnelles, exceptée en ce qui concerne les paramètres de proportions. La formule suivante est obtenue pour ces paramètres (cf. Annexe B) :

$$\pi_k^{q+1} = \sum_{i=1}^N \frac{\mu_{ik}^q}{N} \tag{3.11}$$

Cela correspond donc à un simple calcul pondéré. Concernant les autres paramètres, ils dépendent des densités conditionnelles choisies au départ et sont donc à étudier au cas par cas.

3.5 Conclusion

Ce chapitre a permis d'introduire les principales notions mathématiques utilisées tout au long de la thèse, à savoir : l'approche discriminative/générative, l'approche bayésienne et le classifieur bayésien, les modèles de mélange, et enfin l'algorithme EM. Ces notions représentent la base des modèles de contribution développés dans les trois prochains chapitres de cette thèse. Nous invitons donc le lecteur à s'y référer dès lors qu'il en ressentira le besoin.
Le prochain chapitre introduit la première contribution de la thèse, à savoir le modèle Ignore.

Chapitre 4

Classification Supervisée en Présence de Multiples Annotateurs Incertains

Sommaire

Résumé : En classification supervisée, il est souvent très difficile, voire très onéreux, d'obtenir les réels labels pour toutes les instances d'un jeu de données. Or avec le développement récent d'infrastructures telles qu'Internet, de nombreux services de crowdsourcing ont vu le jour, services permettant de récolter des jeux de données annotés rapidement par plusieurs annotateurs. Ainsi l'étape d'annotation est considérablement facilitée. Néanmoins, ces nouveaux services engendrent très souvent des corpus de données bruités et beaucoup moins fiables, puisque les annotateurs peuvent avoir des niveaux de connaissance très hétérogènes. Ce problème est un obstacle important pour l'apprentissage supervisé, et il est important de le prendre en

considération lors de la génération d'un classifieur dans le but de générer un modèle plus stable face à la diversité des annotations.
Ce chapitre présente la première contribution de la thèse, à savoir le modèle Ignore. Ignore est un modèle probabiliste bayésien qui génère un classifieur en présence de multiples annotateurs incertains. La particularité du modèle est qu'il donne la possibilité aux juges d'exprimer leur incertitude, qu'elle soit totale ou partielle, pour chaque label donné. L'incertitude est intégrée dans le modèle à l'aide d'une approche bayésienne, démarche qui nous a paru naturelle et facile à interpréter dans ce contexte. Le classifieur généré estime la performance de chaque annotateur conditionnellement à leur incertitude, et attribue la classe pour une nouvelle instance donnée. 4 modèles Ignore ont été développés, suivant que l'on se trouve dans un contexte de classification binaire ou multiclasses, et suivant que l'incertitude des annotateurs soit totale ou partielle. Les multiples expérimentations effectuées sur de nombreux corpus de L'UCI Machine Learning Repository valident la performance et la stabilité de Ignore, comparé à des modèles n'intégrant pas l'incertitude des annotateurs. Ignore binaire avec incertitude totale et Ignore multiclasses avec incertitude partielle ont été respectivement publiés dans [Wolley and Quafafou, 2012b, Wolley and Quafafou, 2013a].

4.1 Ignore Binaire avec Incertitude Totale

4.1.1 Formulation du Problème et Notations

Soient N instances $\{x_1, ..., x_N\}$ où chaque exemple x_i est décrit par D descripteurs et étiqueté par T annotateurs experts ou non. Soit y_i^t le label assigné à l'instance x_i par l'annotateur t. Dans un contexte de classification binaire, seules deux classes $\{0, 1\}$ sont disponibles dans le jeu de données. Par ailleurs, on donne la possibilité aux juges d'exprimer leur incertitude. On se place alors dans un cadre d'incertitude ou de connaissance totale. En cas d'incertitude totale (ignorance), les annotateurs ajoutent le caractère '?' au label. Ainsi, alors que l'on a habituellement $y_i^t \in \mathcal{Y} = \{0, 1\}$, on se retrouve dans ce contexte avec $y_i^t \in \mathcal{Y} = \{0, 1\} \cup \{(k, ?)\}_{k=0}^{1}$, où les labels $\{(k, ?)\}_{k=0}^{1}$ représentent les labels dans les situations incertaines.
Soit z_i le véritable label pour la i-ème instance. z_i est inconnu, et $z_i \in \mathcal{Z} = \{0, 1\} \subset \mathcal{Y}$, avec $\mathcal{Z}$ l'espace des véritables labels. On note alors les matrices $X = \left[x_1^T; ...; x_N^T\right] \in R^{N \times D}$ (où x^T désigne la matrice transposée de x), la matrice composée des instances en lignes et des descripteurs en colonnes, $Y = \left[y_1^{(1)}, ..., y_1^{(T)}; ...; y_N^{(1)}, ..., y_N^{(T)}\right] \in R^{N \times T}$ la matrice composée des annotations des juges, et $Z = [z_1, ..., z_N]^T$ le vecteur des réels labels (inconnu).

Conditionnellement à l'incertitude de chaque annotateur, notre objectif est :
- d'estimer le vecteur $Z = [z_1, ..., z_N]^T$ des véritables labels pour toutes les instances du jeu de données initial,
- de produire un classifieur pour prédire le véritable label z pour une nouvelle instance x,

– d'estimer la performance de chaque annotateur, dans les situations de doute et de connaissance.

Une question se pose alors tout naturellement : de quelle manière prendre en compte les labels incertains $\{(k, ?)\}_{k=0}^{1}$ dans le modèle ? Comment différencier les labels incertains des labels certains ? En d'autres termes, comment peut-on modéliser l'incertitude exprimée par chaque annotateur ?

On commence par définir une matrice d'incertitude $H \in R^{N \times T}$ comme suit :

$$h_i^t = \begin{cases} 0 & \text{si } y_i^t = \{0, 1\} \\ 1 & \text{sinon.} \end{cases} \tag{4.1}$$

H est une matrice binaire reflétant l'incertitude de chaque annotateur pour chaque label donné, et permet ainsi d'obtenir une traçabilité de cette dernière. Une fois H définie, on revient à un problème de classification binaire classique en présence de multiples annotateurs, avec, en supplément, la matrice H d'incertitude.
La prochaine section présente les détails de la modélisation du problème.

4.1.2 Modélisation du Problème

Le problème de classification supervisée, où l'on cherche à prédire une variable catégorielle, est souvent ramené au problème de l'estimation de la loi conditionnelle $P(Z|X)$, probabilité d'appartenance aux classes sachant les observations. En présence de multiples annotateurs, la loi conditionnelle à estimer devient alors $P(Z|X, Y)$, avec Y la matrice des labels des juges. Cette probabilité a été très largement étudiée et estimée dans de précédents travaux [Raykar et al., 2010, Raykar and Yu, 2011]. A la différence des méthodes déjà développées, on souhaite ici aller plus loin en incluant, en plus des observations des annotateurs, leur incertitude représentée par la matrice H. Ainsi, le modèle que nous étudions consiste à estimer la loi conditionnelle $P(Z|X, Y, H)$.
Deux types d'approches peuvent être envisagées pour répondre à ce problème : les méthodes discriminatives et les méthodes génératives (cf Section 3.1). En l'absence d'expression explicite pour $P(Z|X, Y, H)$, nous optons pour une méthode générative, où l'on estime pour commencer la loi jointe $P(X, Y, Z|H)$, la distribution conditionnelle de Z pouvant alors être déduite par application de la règle de Bayes. Afin d'estimer les paramètres du modèles (définis plus tard), la méthode générative va maximiser la vraisemblance de la distribution conditionnelle jointe $P(X, Y, Z|H)$. On construit alors un modèle probabiliste qui décrit les relations entre les variables x, y, z, et h. La structure du modèle peut être vue à la Figure 4.1.

Le but ici est d'estimer la probabilité $P(Z|X, Y, H)$, en d'autres termes la probabilité d'obtenir les réels labels Z conditionnellement aux descripteurs X, aux labels Y et à l'incertitude H. Afin d'estimer cette probabilité, nous procédons de la façon suivante :

1. On s'intéresse tout d'abord à la probabilité conditionnelle jointe $P(X, Y, Z|H)$, dont la maximisation de la vraisemblance a posteriori engendre une estimation des

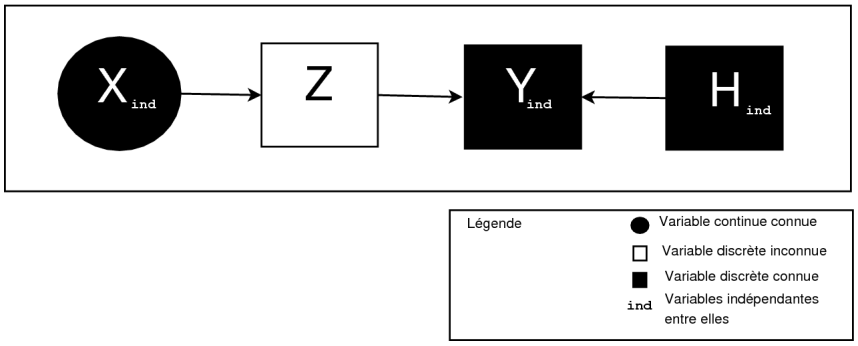


FIGURE 4.1 – Structure du Modèle Ignore

paramètres du modèle qui permettront d'estimer les deux probabilités $P(Z|X,H)$ et $P(Y|X,Z,H)$,

2. Une fois ces probabilités estimées, l'utilisation de la règle de Bayes nous permet de calculer $P(Z|X,Y,H)$, puisque nous avons $P(Z|X,Y,H) = \frac{P(Y|X,Z,H)}{P(Z|X,H)}$,
3. Les labels Z étant inconnus, l'étape d'estimation des paramètres fera appel à une combinaison de l'algorithme EM (Expectation-Maximisation algorithm), de la méthode de Newton-Raphson et de la méthode LBFGS quasi-Newton (cf. Section 3.4).

4.1.3 Estimateur de Maximum a Posteriori

Le but pour commencer est de calculer la probabilité jointe $P(X,Y,Z|H)$. Soient Θ les différents paramètres à estimer (qui seront précisés au fur à et mesure du développement du modèle). En supposant que les instances sont indépendantes entre elles, le maximum de vraisemblance suivant les paramètres Θ peut s'ecrire :

$$P(X,Y,Z|H,\Theta) = \prod_{i=1}^{N} P[x_i, y_i^1, ..., y_i^T, z_i | h_i^1, ..., h_i^T, \Theta]$$

En considérant le théorème de Bayes, on a :

$$P(X,Y,Z|H,\Theta) = \prod_{i=1}^{N}\prod_{t=1}^{T} P[y_i^t, z_i | x_i, h_i^t, \Theta] P[x_i | h_i^t, \Theta] \tag{4.2}$$

$$\propto \prod_{i=1}^{N}\prod_{t=1}^{T} P[y_i^t, z_i | x_i, h_i^t, \Theta] \tag{4.3}$$

Il n'est pas nécessaire ici de considérer la distribution de X puisqu'elle ne dépend pas de Z ou de Y. Or notre objectif est de modéliser ces deux distributions.
Dans un contexte de classification binaire, les réels labels Z peuvent uniquement prendre

les valeurs 0 ou 1. En décomposant $P[X, Y, Z|H, \Theta]$ suivant ces valeurs, on peut écrire :

$$P[X, Y, Z|H, \Theta] = \prod_{i=1}^{N} P(y_i^1, ..., y_i^T | z_i = 1, h_i^1, ..., h_i^T, \Theta_{y_1}) P(z_i = 1 | x_i, \Theta_{z_1}) + \prod_{i=1}^{N} P(y_i^1, ..., y_i^T | z_i = 0, h_i^1, ..., h_i^T, \Theta_{y_0}) P(z_i = 0 | x_i, \Theta_{z_0})$$

où $\Theta = \{\Theta_{y_0}, \Theta_{y_1}, \Theta_{z_0}, \Theta_{z_1}\}$.

Or $P(z_i = 0|x_i, \Theta_{z_0}) = 1 - P(z_i = 1|x_i, \Theta_{z_1})$. On note alors $\Theta_{z_0} = \Theta_{z_1} = \Theta_z$, et par conséquent $\Theta = \{\Theta_{y_0}, \Theta_{y_1}, \Theta_z\}$. On étudie chaque distribution conditionnelle individuellement.

Concernant $P(z_i = 1|x_i, \Theta_z)$, étant dans un contexte de classification binaire, on opte pour la régression logistique, distribution très souvent utilisée pour son efficacité, sa simplicité, et sa facilité d'utilisation. Ainsi, on a :

$$P[z_i = 1|x_i, w] = \frac{1}{1 + e^{-w^T x_i}} \tag{4.4}$$

où $w \in R^d$.
Les paramètres à estimer pour la distribution de z étant dans ce contexte le vecteur w, on a $\Theta_z = \{w\}$.

Concernant $P(y_i^1, ..., y_i^T | z_i = 1, h_i^1, ..., h_i^T, \Theta_{y_1})$, en supposant que les annotateurs (resp. leur incertitude) sont indépendants (resp. indépendantes) entre eux (resp. elles), on obtient :

$$P[y_i^1, ..., y_i^T | z_i = 1, h_i^1, ..., h_i^T, \Theta_{y_1}] = \prod_{t=1}^{T} P(y_i^t | z_i = 1, h_i^t, \Theta_{y_1}^t)$$

On modélise y par un mélange de deux distributions de Bernoulli, suivant la valeur prise par h :

$$y_i^t \sim \begin{cases} Ber(\alpha_0^t) & \text{si } h_i^t = 0 \\ Ber(\alpha_1^t) & \text{sinon.} \end{cases} \tag{4.5}$$

avec $\alpha_0 = P[y_i^t = 1|z_i = 1, h_i^t = 0]$ et $\alpha_1 = P[y_i^t = 1|z_i = 1, h_i^t = 1]$. On définit alors le modèle de mélange suivant :

Definition 6. *Soit $y_i = \left\{y_i^1, y_i^2, ..., y_i^T\right\}$ le vecteur de labels donné par les T annotateurs pour la i-ème instance. Dans le cas où $z_i = 0$, y_i est modélisée par le modèle de mélange de deux distributions de Bernoulli, de paramètres $\alpha_0 = \left\{\alpha_0^1, \alpha_0^2, ..., \alpha_0^T\right\}$ et $\alpha_1 = \left\{\alpha_1^1, \alpha_1^2, ..., \alpha_1^T\right\}$, suivant l'incertitude h des annotateurs :*

$$y_i^t \sim h_i^t * Ber(\alpha_1^t) + (1 - h_i^t) * Ber(\alpha_0^t)$$

où α_0^t (resp. α_1^t) représente le paramètre de la loi Bernoulli pour les labels certains (resp. incertains).

Par conséquent, les paramètres à estimer pour la distribution de y_i dans le cas où $z_i = 0$ sont $\Theta_{y_1} = \{\alpha_0, \alpha_1\}$. On peut alors écrire :

$$P[y_i^1, ..., y_i^T | z_i = 1, h_i^1, ..., h_i^T, \Theta_{y_1}] = \prod_{t=1}^{T} \left[h_i^t (\alpha_1^t)^{y_i^t} (1 - \alpha_1^t)^{1-y_i^t} + (1 - h_i^t)(\alpha_0^t)^{y_i^t} (1 - \alpha_0^t)^{1-y_i^t} \right] \quad (4.6)$$

Respectivement, pour $z_i = 0$, on pose $\Theta_{y_0} = \{\beta_0, \beta_1\}$, avec $\beta_0 = P[y_i^t = 1 | z_i = 0, h_i^t = 0]$ et $\beta_1 = P[y_i^t = 1 | z_i = 0, h_i^t = 1]$. On peut alors écrire :

$$P[y_i^1, ..., y_i^T | z_i = 0, h_i^1, ..., h_i^T, \Theta_{y_0}] = \prod_{t=1}^{T} h_i^t (\beta_1^t)^{1-y_i^t} (1 - \beta_1^t)^{y_i^t} + (1 - h_i^t)(\beta_0^t)^{1-y_i^t} (1 - \beta_0^t)^{y_i^t} \quad (4.7)$$

Les distributions à présent fixées, les paramètres du modèle à estimer sont finalement $\Theta = \{w, \alpha_0, \beta_0, \alpha_1, \beta_1\}$. En posant $a_i = Pr[y_i^1, ..., y_i^T | z_i = 1, h_i^1, ..., h_i^T, \Theta_{y_1}]$, $b_i = Pr[y_i^1, ..., y_i^T | z_i = 0, h_i^1, ..., h_i^T, \Theta_{y_0}]$ et $p_i = Pr[z_i = 1 | x_i, w]$, la vraisemblance peut s'écrire :

$$P(X, Y, Z | H, \Theta) = \prod_{i=1}^{N} [a_i p_i + b_i (1 - p_i)] \quad (4.8)$$

Notre but à présent est d'estimer les différents paramètres Θ. Pour cela, le fait de connaitre le domaine d'incertitude de chaque annotateur nous permet d'avoir recours à une méthode bayésienne et de fixer des a prioris sur chacun des paramètres. Notre but est alors de maximiser la vraisemblance a posteriori à l'aide de l'Estimateur du Maximum a Posteriori (MAP) (cf. Section 3.4.4) :

$$\hat{\Theta}_{MAP} = argmax_{\Theta} \{ln\ P[X, Y, Z | H, \Theta] + ln\ P[\Theta]\} \quad (4.9)$$

Différents a prioris peuvent être choisis pour chaque paramètre. La section qui suit décrit les a prioris utilisés dans notre travail.

4.1.4 Distributions a Priori

D'un point de vue général, l'analyse statistique bayésienne vise à exploiter le plus efficacement possible l'information apportée par les observations sur le paramètre à estimer, pour ensuite construire des procédures d'inférence sur ces derniers. Dans le paradigme bayésien, le paramètre à estimer n'est plus considéré comme inconnu, mais comme une variable aléatoire où on considère que l'incertitude sur le paramètre peut être décrite par une distribution de probabilité π sur ce même paramètre. Cette distribution est appelée distribution a priori (par opposition à la distribution a posteriori qui inclut les informations contenues dans les observations). L'avantage de l'utilisation d'une distribution a

priori est qu'elle représente la manière la plus efficace de résumer l'information disponible (ou le manque d'information) sur ce paramètre. Un autre avantage d'un point de vue plus technique est qu'elle permet de construire une approche mathématiquement justifiée tout en opérant conditionnellement aux observations et en restant dans un schéma probabiliste. Des arguments de natures différentes sur l'optimalité de cet outil sont fournis dans [Bernardo and Smith, 1994] et [Robert, 2007]. Cependant, d'un point de vue plus pratique, il reste souvent difficile de choisir la distribution a priori qui convient, surtout qu'il a été constaté que le choix de l'a priori conduit à des inférences significativement divergentes. Il existe néanmoins des lois calibrées, en fonction de la distribution des observations, dites lois conjuguées, et des lois à faible contenu informatif, dites lois non-informatives. On renvoie le lecteur à la Section 3.2.2 pour plus d'informations.

Ainsi, le choix des lois a priori est une étape fondamentale dans l'analyse bayésienne. Dans notre cas, le choix des a prioris supposés sur les paramètres dépend essentiellement de la matrice H d'incertitude des juges. En effet, on souhaite ici que notre modèle Ignore favorise les labels sûrs en imposant un a priori fort sur ces derniers, et qu'il accorde, a contrario, beaucoup moins de poids aux labels incertains lors de la génération du classifieur. On précise alors ci-dessous les lois a priori fixées pour différents paramètres $\Theta = \{w, \alpha_0, \beta_0, \alpha_1, \beta_1\}$.

A Priori sur les Paramètres $\{\alpha_0, \beta_0\}$

D'après les équations (4.6) et (4.7), on a $\alpha_0^t = P\left[y_i^t = 1 | z_i = 1, h_i^t = 0\right]$ et $\beta_0^t = P\left[y_i^t = 0 | z_i = 0, h_i^t = 0\right]$. En d'autres termes, α_0^t et β_0^t représentent respectivement la sensibilité (taux de vrais positifs) et la spécificité (1-taux de faux positifs) de l'annotateur t dans les cas de situations sûres. Or ces paramètres représentent la probabilité d'un évènement binaire et peuvent donc suivre une distribution de Bernoulli. Un choix naturel de distribution a priori est alors la loi Beta de paramètre (a,b) (a et b strictement positif), représentant la loi a priori conjuguée de la loi de Bernoulli. On suppose alors un a priori Beta sur les deux paramètres $\{\alpha_0, \beta_0\}$, c'est-à-dire :

$$\begin{cases} P(\alpha_0^t | a_{01}^t, a_{02}^t) \sim Beta(\alpha_0^t | a_{01}^t, a_{02}^t) \\ P(\beta_0^t | b_{01}^t, b_{02}^t) \sim Beta(\beta_0^t | b_{01}^t, b_{02}^t) \end{cases}$$

Les paramètres $\{a_{01}^t, a_{02}^t, b_{01}^t, b_{02}^t\}$ sont alors estimés comme suit : Soient (a,b) les paramètres de la distribution beta, (a,b) sont tous deux estimés en résolvant le système d'équations suivant :

$$\mu = a/(a+b) \tag{4.10}$$

$$\int_{u1}^{u2} Beta(p|a,b)dp = C \tag{4.11}$$

où p est la probabilité à calculer, dans notre cas la sensibilité et la spécificité. μ est la moyenne de la distribution Beta, C, u1 et u2 sont trois constantes à initialiser.
Dans notre cas, μ représente la probabilité en moyenne pour que la sensibilité ou la

spécificité soient vraies. En d'autres termes, μ représente la probabilité en moyenne que les annotateurs donnent la bonne réponse. Or dans un contexte de certitude, on considère μ proche de 1, et on fixe $\mu = 0.9$.
En ce qui concerne l'équation (4.11), elle définit le pourcentage de chance que p se situe dans l'intervalle [u1,u2]. Dans un contexte toujours de certitude où $h = 0$, on pose $P[\alpha_0^t \in [0.5 : 1]] = 0.9$, et $P[\beta_0^t \in [0.5 : 1]] = 0.9$. En d'autres termes, on suppose que la probabilité pour que la sensibilité et la spécificité soient compris dans l'intervalle [0.5,1] est de 0.9. Cela reflète alors la confiance accordée aux labels des juges.
Par la suite, nous fixons les valeurs des paramètres ci-dessus de la façon suivante : C = 0.9 et [u1,u2] = [0.5,1].

A Priori sur les Paramètres $\{\alpha_1, \beta_1\}$

D'après les équations (4.6) et (4.7), on a $\alpha_1^t = P\left[y_i^t = 1 | z_i = 1, h_i^t = 1\right]$ et $\beta_1^t = P\left[y_i^t = 0 | z_i = 0, h_i^t = 1\right]$. On se place ici dans le cas où les annotateurs sont supposés être totalement ignorants. On suppose alors que dans ce cas, les juges annotent les instances aléatoirement. En d'autres termes, $\alpha_1^t = P(random(0,1) = 1 | z_i = 1, h_i^t = 1)$ et $\beta_1^t = P(random(0,1) = 0 | z_i = 0, h_i^t = 1)$, où *random* correspond à la fonction aléatoire. Trois différents a priori sont étudiés :

- A priori Beta(a,b) : Comme α_1^t et β_1^t représentent encore une fois la probabilité d'un évènement binaire, la distribution Beta B(a,b) est supposée sur les deux paramètres :
$$\begin{cases} P(\alpha_1^t | a_{11}^t, a_{12}^t) \sim Beta(\alpha_1^t | a_{11}^t, a_{12}^t) \\ P(\beta_1^t | b_{11}^t, b_{12}^t) \sim Beta(\beta_1^t | b_{11}^t, b_{12}^t) \end{cases}$$
Les paramètres $\{a_{01}^t, a_{02}^t, b_{01}^t, b_{02}^t\}$ sont de nouveau estimés à l'aide des équations (4.10) et (4.11). Dans un contexte d'ignorance totale, on suppose qu'en moyenne, la probabilité pour que les annotateurs donnent la bonne réponse est de 0.5. Ainsi, on pose $\mu = 0.5$.
Concernant les paramètres u1, u2 et C, on suppose que la probabilité pour que la sensibilité et la spécificité des annotateurs soient comprises entre 0.4 et 0.5 est de 0.9, ce qui s'écrit $P[\alpha_1^t \in [0.4 : 0.6]] = 0.9$ (resp. $P[\beta_1^t \in [0.4 : 0.6]] = 0.9$). Dans ces conditions, on a C = 0.9 et [u1,u2] = [0.4,0.5].

- A priori Beta(1,1) : les a prioris non informatifs sont souvent utilisés pour modéliser l'incertitude, puisqu'ils reflètent le manque de connaissance sur un paramètre (cf Section 3.2.2). Dans le cas d'un évènement binaire, plusieurs travaux ont utilisé la loi a priori uniforme U[0,1] en tant que distribution non informative [Bernardo and Smith, 1994, Kass and Wasserman, 1996]. En effet, d'après Bayes et Laplace : *quand aucun renseignement n'est connu sur* Θ *en avance, alors il faut faire en sorte que l'a priori* $\pi(\Theta)$ *soit uniforme, c'est-à-dire que toutes les sorties pour* Θ *soient possibles avec la même probabilité.* [1] Or cette distribution équivaut à

1. when nothing is known about Θ in advance, let the prior $\pi(\Theta)$ be a uniform prior, that is, all

la distribution Beta B(1,1). En effet, la loi Beta(a,b) a pour densité de probabilité :

$$P(x) = \begin{cases} \frac{x^{a-1}(1-x)^{b-1}}{\int_0^1 \frac{t^{a-1}}{(1+t)^{a+b}}dt} & \text{si } x \in [0,1] \\ 0 & \text{sinon} \end{cases}$$

Or en remplaçant les valeurs de a et de b par 1, on retrouve la densité d'une loi uniforme.
Un a priori Beta(1,1) sur les paramètres α_1^t et β_1^t supposent alors que toutes les valeurs sont possibles entre 0 et 1, et ce, de manière équiprobable. Cet a priori peut être écrit comme suit :

$$\begin{cases} P(\alpha_1^t|1,1) = Beta(\alpha_1^t|1,1) \\ P(\beta_1^t|1,1) = Beta(\beta_1^t|1,1) \end{cases}$$

– A priori Jeffreys : Un autre a priori non informatif très souvent utilisé est l'a priori de Jeffreys [Jeffreys, 1946] (cf Section 3.2.2). De nombreux travaux ont utilisé cet a priori [Figueiredo and Nowak, 2001, Figueiredo, 2002], ses propriétés permettant d'y faire appel dans de nombreuses applications. Il se calcule de la façon suivante : Soit Θ le vecteur des paramètres à estimer, l'a priori de Jeffreys p est :

$$p(\Theta) \propto \sqrt{det(I(\Theta))}$$

où I correspond à l'information de Fisher. Ainsi, dans le cas d'un modèle Bernoulli avec γ pour paramètre, on calcule l'apriori de Jeffreys de la façon suivante :

$$P(\gamma) \propto \sqrt{E\left[\left(\frac{d}{d\gamma} log f(x|\gamma)\right)^2\right]} \tag{4.12}$$

$$\propto \sqrt{\gamma\left(\frac{1}{\gamma} - \frac{0}{1-\gamma}\right)^2 + (1-\gamma)\left(\frac{0}{\gamma} - \frac{1}{1-\gamma}\right)^2} \tag{4.13}$$

$$\propto \frac{1}{\sqrt{\gamma(1-\gamma)}} \tag{4.14}$$

On a alors pour (α_1^t, β_1^t) :

$$\begin{cases} P(\alpha_1^t) = & \frac{1}{\sqrt{\alpha_1^t(1-\alpha_1^t)}} \\ P(\beta_1^t) = & \frac{1}{\sqrt{\beta_1^t(1-\beta_1^t)}} \end{cases}$$

Ce qui correspond en réalité la loi Beta(a,b) de paramètres a = b = 0.5.

possible outcomes of Θ have the same probability.

A Priori sur le Paramètre w

Par souci de simplification, on suppose un a priori Gaussien sur les poids w de telle sorte que $w \sim N(w|0,1)$.

A présent que les distributions a priori sont fixées, il nous faut maximiser la vraisemblance a posteriori et exhiber l'algorithme Ignore. Ce sera l'objectif de la section qui suit.

4.1.5 Algorithme IGNORE

Cette section a pour premier objectif l'estimation des paramètres Θ du modèle. Pour cela, on maximise la vraisemblance a posteriori afin d'obtenir une estimation du maximum a posteriori $\hat{\Theta}_{MAP}$ définie par l'équation (4.9). Les variables latentes z étant manquantes, une méthode très utilisée est la maximisation de vraisemblance a posteriori est l'algorithme EM (Expectation Maximisation algorithm), algorithme adapté à l'estimation du MAP. Chaque itération de l'algorithme EM consiste en 2 étapes E et M. Dans l'étape E (expectation-step), les valeurs manquantes sont estimés à l'aide des données observées et de l'estimation ponctuelle des paramètres. Dans l'étape M (Maximisation-step), la vraisemblance est maximisée sous l'hypothèse que les données manquantes sont connues. En adaptant l'algorithme EM à notre problème, on obtient les deux étapes qui suivent :

E-Step : Soient X,Y,H donnés, ainsi que l'estimation ponctuelle des paramètres Θ. Les labels réels z sont estimés par le calcul de $\mu_i = Pr\left[z_i = 1|y_i^1, ..., y_i^T, h_i^1, ..., h_i^T, x_i, \Theta\right]$. D'après le théorème de Bayes, on a :

$$\mu_i \ \propto \ Pr[y_i^1, ..., y_i^T|z_i = 1, h_i^1, ..., h_i^T, \Theta] \times Pr[z_i = 1|x_i, \Theta] \tag{4.15}$$

Et d'après les équations (4.4) et (4.6), on obtient l'expression suivante pour μ_i :

$$\mu_i \ = \ \frac{a_i p_i}{a_i p_i + b_i(1-p_i)} \tag{4.16}$$

M-Step : En se basant sur l'estimation de μ_i obtenue à l'étape E et sur les observations X et Y, les paramètres Θ peuvent à leur tour être estimés en ramenant à zéro l'espérance du log de la vraisemblance suivant chaque paramètre, i.e :

$$E\left[ln\ P\left[X, Y, Z|H, \Theta\right]\right] \ \propto \ ln[P(\Theta)] + \sum_{i=1}^{N} \mu_i ln(p_i a_i) + (1-\mu_i) ln(1-p_i) b_i \tag{4.17}$$

Pour l'a priori Beta(a,b) et Beta(1,1) on obtient pour $j \in \{0,1\}$:

$$\alpha_j^t \ = \ \frac{\sum_{i \in \left\{i|h_i^t = j\right\}} \mu_i y_i^t + a_{j1}^t - 1}{a_{j1}^t + a_{j2}^t - 2 + \sum_{i \in \left\{i|h_i^t = j\right\}} \mu_i} \tag{4.18}$$

$$\beta_j^t = \frac{b_{j1} - 1 + \sum_{i \in \{i|h_i^t=j\}} (1-\mu_i)(1-y_i^t)}{b_{j1}^t + b_{j2}^t - 2 + \sum_{i \in \{i|h_i^t=j\}} (1-\mu_i)} \tag{4.19}$$

Concernant l'a priori de Jeffreys, il est plus difficile de ramener les gradients de α_1^t et β_1^t à 0, les calculs étant peu évidents. On utilise alors l'algorithme LBFGS quasi-Newton [Nocedal and Wright, 2003].
Par souci de clarté, on donne les gradients pour les paramètres $\{\alpha_1^t, \beta_1^t\}$:
On pose $f_{opt} = E\left[ln\ Pr\left[X, Y, Z|H, \Theta\right]\right]$ et on obtient :

$$\frac{\partial f_{opt}}{\partial \alpha_1^t} = \sum_{i=1}^{N} \left[\mu_i(\alpha_1^t - y_i^t) + \frac{1 - 2\alpha_1^t}{2\sqrt{\alpha_1^t(1-\alpha_1^t)}} \right] \tag{4.20}$$

$$\frac{\partial f_{opt}}{\partial \beta_1^t} = \sum_{i=1}^{N} \left[(1-\mu_i)[(1-y_i^t)(1-\beta_1^t) - y_i^t\beta_1^t] + \frac{2\beta_1^t - 1}{2\sqrt{\beta_1^t(1-\beta_1^t)}} \right] \tag{4.21}$$

Enfin, concernant le paramètre w, on utilise la méthode de Newton-Raphson (cf. Section 3.4.2) donnant lieu à :

$$w^{q+1} = w^q - \eta H^{-1} g \tag{4.22}$$

où g est le vecteur gradient, H la matrice hessienne et η le pas. Le vecteur gradient est donné par

$$g(w) = \sum_{i=1}^{N} \left[\mu_i - \tau(w^T x_i)\right] x_i - \Gamma w \tag{4.23}$$

et la matrice hessienne est donnée par :

$$H(w) = -\sum_{i=1}^{N} \left[\tau(w^T x_i)\right] \left[1 - \tau(w^T x_i)\right] x_i x_i^T - \Gamma \tag{4.24}$$

Les deux étapes E et M sont itérées jusqu'à convergence. μ_i est initialisée à $\frac{1}{|P|}\sum_{t=1}^{P} y_i^t$ avec $P = \{t|h_i^t \neq 1\}$.

On résume l'approche par l'Algorithme 5.

Finalement, une fois que les paramètres sont estimés par l'algorithme Ignore, une nouvelle instance x_i est classée en calculant $p(z_i = 1|x_i) = (1 + exp(-w^T x_i))^{-1}$, la probabilité que x_i ait le véritable label z_i égale à 1. A partir de là, on peut facilement déduire la probabilité $p(z_i = 0|x_i)$, puisque l'on a $p(z_i = 0|x_i) = 1 - p(z_i = 1|x_i)$. Le label attribué à x_i sera finalement celui avec la plus haute probabilité.

Nous allons à présent étendre l'approche Ignore développée dans cette partie à des contextes plus généraux, en intégrant l'incertitude partielle et/ou la classification multi-classes. Nous allons voir que ces extensions nécessitent en réalité très peu de changements par rapport au modèle Ignore de base.

Algorithm 5 Algorithme Ignore

1: Données initiales : X, Y, H, seuil ϵ
2: Initialiser : q=0, $\mu, w^q, \alpha_0, \alpha_1, \beta_0, \beta_1, \Gamma$
3: Calculer : g, H, w^{q+1}
4: **while** $||w^{q+1} - w^q||^2 \leq \epsilon$ **do**
5: q = q+1
6: Etape E : Estimer μ à l'aide de l'équation (4.16).
7: Etape M : Calculer $\alpha_0, \beta_0, w^{q+1}, H$ à l'aide des équations (4.18), (4.19), (4.22), (4.23), (4.24). Calculer α_1, β_1 à l'aide des équations (4.18), (4.19) pour les a priori Beta(a,b) et Beta(1,1). Pour l'a priori de Jeffreys, utiliser les équations (4.20) et (4.21).
8: **end while**
9: Retourner les paramètres $(\alpha_0, \alpha_1, \beta_0, \beta_1, w)$

4.2 Extensions du Modèle Ignore

4.2.1 Cas Binaire avec Incertitude Partielle

Le modèle Ignore précédemment développé concerne le cas de l'ignorance totale des juges. Cependant, ce contexte est une vision simplifiée de la réalité puisque souvent, les annotateurs sont partiellement certains des labels attribués. Ainsi, il serait intéressant d'étendre Ignore au contexte d'incertitude partielle.
Dans un tel contexte, on définit un degré d'incertitude compris entre 0 et 1, de telle sorte que plus ce degré est proche de 1, plus le taux d'ignorance est élevé. A l'inverse, plus ce degré est proche de 0, plus le taux d'ignorance est faible. Dans les cas particuliers où ce degré vaut 0 ou 1, on revient au modèle Ignore binaire avec incertitude totale précédemment développé.
Ainsi, dans un contexte d'incertitude partielle, les annotateurs attribuent à chaque label un degré d'incertitude, degré correspondant à l'expression de leur incertitude quant aux labels donnés. On montre dans cette partie que l'extension du modèle Ignore au cadre de l'incertitude partielle est très naturelle et simple.

Formalisation du problème et Notations

La formalisation du problème est ici très similaire au modèle Ignore dans le cas binaire avec incertitude totale. On pose $X \in R^{N \times D}$ la matrice des données, et $Y \in R^{N \times T}$ la matrice d'annotations des juges. Dans un contexte binaire avec incertitude partielle, chaque juge est tenu d'annoter les instances à l'aide des labels 0 ou 1, tout en précisant son degré d'incertitude, degré qui est compris entre 0 et 1 (0 correspondant à une certitude totale, et 1 correspondant à une ignorance totale). On a alors $y_i^t \in \mathcal{Y} = \{(0, u_i^t) \cup (1, u_i^t)\}$ avec $u_i^t \in [0, 1]$. On redéfinit alors la matrice d'incertitude $H \in R^{N \times T}$ dans ce contexte. $\forall i \in \{1, ..., N\}$, $\forall\, t \in \{1, ..., T\}$, on a :

$$h_i^t = u_i^t \tag{4.25}$$

Modélisation du Problème

Conditionnellement à l'incertitude H, l'objectif reste le même que pour le modèle Ignore binaire avec incertitude totale. Formellement, la probabilité à estimer est toujours $P(Z|X,Y,H)$.

Estimateur de Maximum a Posteriori

La principale différence dans la décomposition de $P(Z|X,Y,H)$ réside dans l'interprétation des paramètres $\{\alpha_0,\beta_0,\alpha_1,\beta_1\}$. En effet, alors que $\{\alpha_0,\beta_0\}$ (resp. $\{\alpha_1,\beta_1\}$) représentaient les paramètres du modèle dans le cas de connaissance (resp. d'ignorance), ici cette distinction n'est plus valable puisque le taux d'incertitude h est compris entre [0,1]. Ainsi, ici nous avons $\alpha_0 = \alpha_1 = P[y_i^t = 1|z_i = 1, h_i^t]$ et $\beta_0 = \beta_1 = P[y_i^t = 1|z_i = 0, h_i^t]$. La différence entre α_0 et α_1 (resp. β_0 et β_1) réside dans le choix des a prioris sur ces paramètres. En effet, d'après la définition 6, la distribution de y lorsque le réel label $z_i = 1$ s'écrit :

$$y_i^t \sim h_i^t * Ber(\alpha_1^t) + (1 - h_i^t) * Ber(\alpha_0^t)$$

Le paramètre α_1 est pris en compte pour le taux d'incertitude h_i^t, alors que le paramètre α_0 est pris en compte pour le taux de certitude $(1 - h_i^t)$. Ainsi, un fort a priori sera considéré pour les paramètres α_0 (idem pour β_0), alors qu'un a priori plus faible sera considéré pour les paramètres α_1 (idem pour β_1).

Les différents paramètres du modèle à estimer sont toujours $\Theta = \{w,\alpha_0,\beta_0,\alpha_1,\beta_1\}$. Pour cela, on maximise la vraisemblance a posteriori à l'aide de l'estimateur MAP défini à l'équation 4.9.

Distribution a Priori

Les a prioris considérés pour chaque paramètre sont les mêmes que ceux décrits à la Section 4.1.4.

Algorithme Ignore

Les paramètres sont estimés à l'aide de l'algorithme EM, les étapes E et M pouvant être vues à la Section 4.1.5. A l'étape E, $\mu_i = P\left[z_i = 1|y_i^1,...,y_i^T,h_i^1,...,h_i^T,x_i,\Theta\right]$ est estimée à l'aide de l'équation (4.16). A l'étape M, les paramètres Θ sont estimés en ramenant à zéro l'espérance du log de la vraisemblance définie à l'équation (4.17).
Or les paramètres $\{\alpha_0,\beta_0,\alpha_1,\beta_1\}$ dépendent de la valeur de l'incertitude h qui est dans ce contexte comprise dans l'intervalle $[0,1]$. Ainsi le calcul des gradients donne :

$$\begin{cases} \alpha_0^t = & \dfrac{\sum_i (1-h_i^t)\mu_i y_i^t + a_{j1}^t - 1}{a_{j1}^t + a_{j2}^t - 2 + \sum_i (1-h_i^t)\mu_i} \\ \alpha_1^t = & \dfrac{\sum_i (h_i^t)\mu_i y_i^t + a_{j1}^t - 1}{a_{j1}^t + a_{j2}^t - 2 + \sum_i (h_i^t)\mu_i} \end{cases} \tag{4.26}$$

$$\begin{cases} \beta_0^t = & \dfrac{b_{j1} - 1 + \sum_i (1 - h_i^t)(1 - \mu_i)(1 - y_i^t)}{b_{j1}^t + b_{j2}^t - 2 + \sum_i (1 - h_i^t)(1 - \mu_i)} \\ \beta_1^t = & \dfrac{b_{j1} - 1 + \sum_i h_i^t (1 - \mu_i)(1 - y_i^t)}{b_{j1}^t + b_{j2}^t - 2 + \sum_i h_i^t (1 - \mu_i)} \end{cases} \tag{4.27}$$

Concernant l'a priori de Jeffrey, on obtient :

$$\frac{\partial f_{opt}}{\partial \alpha_1^t} = \sum_{i=1}^{N} h_i^t \left[\mu_i (\alpha_1^t - y_i^t) + \frac{1 - 2\alpha_1^t}{2\sqrt{\alpha_1^t (1 - \alpha_1^t)}} \right] \tag{4.28}$$

$$\frac{\partial f_{opt}}{\partial \beta_1^t} = \sum_{i=1}^{N} h_i^t \left[(1 - \mu_i)[(1 - y_i^t)(1 - \beta_1^t) - y_i^t \beta_1^t] + \frac{2\beta_1^t - 1}{2\sqrt{\beta_1^t (1 - \beta_1^t)}} \right] \tag{4.29}$$

L'algorithme reste alors inchangé (cf Algorithme 5), mais les équations (4.18), (4.19), (4.20), et (4.21) de l'algorithme EM pour l'estimation des paramètres sont remplacées, respectivement, par les équations (4.26), (4.27), (4.28) et (4.29).

La section qui suit est consacrée à l'extension d'Ignore au cas multiclasses avec incertitude totale.

4.2.2 Cas Multiclasses avec Incertitude Totale

Nous allons voir dans cette section que le modèle Ignore est très facilement généralisable au cas de plusieurs classes, la clé étant essentiellement de modifier les distributions binaires attribuées à chaque variable par des distributions appropriées à une classification multiclasses.

Formalisation du Problème et Notations

On pose $X \in R^{N \times D}$ la matrice des données, $Y \in R^{N \times T}$ la matrice d'annotations des juges. On revient au cadre de l'incertitude totale, où les annotateurs attribuent le symbole '?' en plus du label en cas d'ignorance. Ainsi, pour (K+1) classes disponibles, $y_i^t \in \mathcal{Y} = \{0, 1, ..., K\} \cup \{(k, ?)\}_{k=0}^{K}$, et le réel label z_i appartient à $\mathcal{Z} = \{0, 1, ..., K\}$. De plus, dans ce contexte, la matrice H d'incertitude est définie comme suit :

$$h_i^t = \begin{cases} 0 & \text{si } y_i^t = \{0, 1, ..., K\} \\ 1 & \text{sinon.} \end{cases} \tag{4.30}$$

Modélisation du Problème

Conditionnellement à l'incertitude H, l'objectif est toujours d'estimer la probabilité $P(Z|X, Y, H)$. Cependant, le contexte n'étant plus une classification binaire, la principale modification de ce modèle concerne les distributions adoptées pour chaque variable. C'est ce que nous allons étudier ci-dessous.

Estimateur de Maximum a Posteriori

La décomposition de la probabilité $P(X, Y, Z|H, \Theta)$ dans le cas multiclasses donne :

$$P(X, Y, Z|H, \Theta) = \prod_{i=1}^{N} \prod_{t=1}^{T} P[y_i^t, z_i|x_i, h_i^t, \Theta] P[x_i|h_i^t, \Theta] \tag{4.31}$$

$$\propto \prod_{i=1}^{N} \prod_{t=1}^{T} P[y_i^t, z_i|x_i, h_i^t, \Theta] \tag{4.32}$$

Or le label réel z appartient à $\mathcal{Z} = \{0, 1, ..., K\}$. En décomposant suivant les différentes valeurs possibles pour z, la vraisemblance peut s'écrire comme suit :

$$P[X, Y, Z|H, \Theta] \propto \prod_{i=1}^{N} \sum_{k=0}^{K} P(z_i = k|x_i, \Theta_{z_k}) \prod_{t=1}^{T} P(y_i^t|z_i = k, h_i^t, \Theta_{y_k}^t) \tag{4.33}$$

On doit alors estimer une nouvelle fois chaque distribution individuellement.

Concernant $P(z_i = k|x_i, \Theta_{z_k})$, une généralisation de la régression logistique utilisée dans le cas binaire est la loi logistique multinomiale (cf Annexe D.3), qui donne :

$$Pr(z_i = k|x_i, \Theta_{z_k}) = \begin{cases} \frac{1}{1+\sum_{k=0}^{K-1} e^{w_k^T x_i}} & \text{si } k = K \\ \frac{e^{w_k^T x_i}}{1+\sum_{k=0}^{K-1} e^{w_k^T x_i}} & \text{sinon} \end{cases} \tag{4.34}$$

Ainsi on a $\Theta_{z_k} = \{w_k\}\,,\ \forall k \in \{0, 1, ..., K\}$

Concernant $P(y_i^t|z_i = k, h_i^t, \Theta_{y_k}^t)$, on remplace le mélange de Bernoulli par un mélange de distributions multinomiales, représentant la généralisation de la loi de Bernoulli au cas multiclasses. Ainsi, on introduit les deux vecteurs $\alpha_k^t = (\alpha_{k0}^t, ..., \alpha_{kK}^t)$ et $\beta_k^t = (\beta_{k0}^t, ..., \beta_{kK}^t)$ représentant les paramètres de la loi, et on a :

$$y_i^t \sim \begin{cases} Mu(\alpha_k^t) & \text{si } h_i^t = 0 \\ Mu(\beta_k^t) & \text{sinon} \end{cases}$$

avec $\sum_{c=0}^{K} \alpha_{kc}^t = 1$ et $\sum_{c=0}^{K} \beta_{kc}^t = 1$, les α_{kc}^t (resp. β_{kc}^t) dénotant la probabilité que l'annotateur t assigne la classe c à l'instance dont le réel label est k, dans le cas de connaissance (resp. d'incertitude).

Finalement, $\forall k \in \{0, 1, ..., K\}\,, \forall t \in \{1, 2, ..., T\}$, on a $\Theta_{y_k}^t = \{\alpha_k^t, \beta_k^t\}$. On définit alors la distribution de y ci-dessous :

Definition 7. *Soit $y_i = \left\{y_i^1, y_i^2, ..., y_i^T\right\}$ le vecteur de labels donné par les T annotateurs pour la i-ème instance. y_i est modélisée par le mélange de deux distributions Multinomiales de paramètres $\alpha_k^t = (\alpha_{k0}^t, ..., \alpha_{kK}^t)$ et $\beta_k^t = (\beta_{k0}^t, ..., \beta_{kK}^t)$ suivant l'incertitude h*

des annotateurs, ce qui donne :

$$y_i^t \sim (1 - h_i^t) \prod_{c=0}^{K} (\alpha_{kc}^t)^{\delta(y_i^t, c)} + h_i^t \prod_{c=0}^{K} (\beta_{kc}^t)^{\delta(y_i^t, c)}$$

où $\delta(u, v) = 1$ *si* $u = v$, *0 sinon.*

On pose :

$$p_i^k = P(z_i = k | x_i, \Theta_{z_k}) \tag{4.35}$$

$$a_i^k = \prod_{t=1}^{T} P(y_i^t | z_i = k, h_i^t, \Theta_{y_k}^t) \tag{4.36}$$

Finalement, d'après les équations (4.33), (4.35) et (4.36), la vraisemblance peut s'écrire :

$$P(X, Y, Z | H, \Theta) \propto \prod_{i=1}^{N} \sum_{k=0}^{K} p_i^k a_i^k \tag{4.37}$$

où $\Theta = \left\{\Theta_{z_k}, \Theta_{y_k}^t\right\} = \{w_k, \alpha_k^t, \beta_k^t\}$.

Le but est toujours de maximiser la vraisemblance a posteriori à l'aide de l'Estimateur du Maximum a Posteriori (MAP) :

$$\hat{\Theta}_{MAP} = argmax_{\Theta} \{ln\ P[X, Y, Z | H, \Theta] + ln\ P[\Theta]\} \tag{4.38}$$

Distributions a Priori

Dans le cas binaire, la loi Beta avait été optée comme a priori sur les paramètres puisque qu'elle représente la loi conjuguée de la loi de Bernoulli. Dans le cas multiclasses, on opte pour la loi de Dirichlet, représentant à son tour l'a priori conjugué de la loi multinomiale. On doit alors fixer les paramètres de la loi de Dirichlet selon que les labels soient certains ou non.

– **A priori sur les labels certains :**
Dans le cas de connaissance, les paramètres à prendre en considération sont les paramètres $\{\alpha_{kc}^t\}$ de la distribution multinomiale. On suppose alors une distribution de Dirichlet ayant pour paramètres $\{\gamma_{kc}^t\}$ sur l'ensemble des paramètres $\{\alpha_{kc}^t\}$, et on a :

$$\alpha_k^t = (\alpha_{k0}^t, \alpha_{k1}^t, ..., \alpha_{kK}^t) \sim Dir(\gamma_{k0}^t, \gamma_{k1}^t, ..., \gamma_{kK}^t)$$

Les paramètres $\{\gamma_{kc}^t\}$ sont estimés comme suit :

Soit $\lambda = \{\lambda_1, \lambda_2, ..., \lambda_N\}$ les paramètres de la distribution de Dirichlet pour la variable $X = \{X_1, X_2, ..., X_N\}$. λ est estimé en choisissant, selon le contexte dans lequel on se trouve, la moyenne et la variance de la distribution. Une fois ces paramètres choisis, on résout le système suivant :

$$E[X_i] = \frac{\lambda_i}{\sum_{i=1}^{N} \lambda_i} \tag{4.39}$$

$$Var[X_i] = \frac{\lambda_i(\sum_{i=1}^{N} \lambda_i - \lambda_i)}{(\sum_{i=1}^{N} \lambda_i)^2(\sum_{i=1}^{N} \lambda_i + 1)} \tag{4.40}$$

En résolvant ces équations suivant les paramètres λ_i, on obtient $\lambda_i = E[X_i] \sum_{i=1}^{N} \lambda_i$ avec $\sum_{i=1}^{N} \lambda_i = \frac{E[X_i](1-E[X_i])}{Var[X_i]} - 1$.

Dans notre cas, les paramètres $\{\alpha_{kc}^t\}$ représentent les paramètres de la distribution multinomiale dans le cadre de certitude. Ainsi, on suppose que dans un tel contexte, les annotateurs ont de fortes chances d'attribuer le bon label à l'instance, ce que l'on traduit par $E[\alpha_{kk}^t] = 0.9$. En d'autres termes, on suppose que la probabilité pour que l'annotateur t donne le bon label est de 0.9. Concernant les autres paramètres α_{kc}, $\forall\ k \neq c$, on a $\sum_{c=0}^{K} \alpha_{kc}^t = 1$, entrainant alors que $E[\alpha_{kc}^t] = 0.1/K,\ \forall\ k \neq c$.
Enfin, pour la variance, on suppose que $Var[\alpha_{kc}^t] = 0.01$ afin de refléter une forte confiance sur l'a priori posé.
Une fois la moyenne et la variance fixées, les équations (4.39) et (4.40) sont utilisées pour estimer les paramètres γ_{kc}^t.

– **A priori sur les labels incertains :**
Les paramètres à estimer dans le modèle sont les paramètres $\{\beta_{kc}^t\}$ de la loi multinomiale. Comme pour le cas binaire, on considère ici trois différents a priori moins informatifs :

A priori de Dirichlet : On suppose

$$\beta_k^t = (\beta_{k0}^t, \beta_{k1}^t, ..., \beta_{kK}^t) \sim Dir(\phi_{k0}^t, \phi_{k1}^t, ..., \phi_{kK}^t)$$

Or nous sommes dans le cas où $h^t = 1$, c'est-à-dire dans le cas où les annotateurs ignorent le label à donner. On suppose que dans un tel contexte il y a équiprobabilité entre tous les labels. Par conséquent, on fixe $E[\beta_{kc}^t] = 1/(K+1)$, $\forall\ \{k, c\} \in \{0, 1, 2, ..., K\}$. Concernant la variance, on fixe $Var[\beta_{kc}^t] = 0.01$ afin de refléter une nouvelle fois une forte confiance sur l'a priori choisi. Une fois la moyenne et la variance fixées, les équations (4.39) and (4.40) sont à nouveau utilisées pour estimer $\{\phi_k^t\}$.

A priori uniforme : Comme pour le cas binaire, un a priori non informatif souvent utilisé est l'a priori uniforme, qui peut aussi être écrit sous forme de distribution de Dirichlet :

$$\beta_{kc}^t \sim Dir(1)$$

.
En effet, la loi de Dirichlet de paramètres $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_K\}$ admet pour densité

de probabilité :

$$f(x_1, x_2, ..., x_N) = C \prod_{i=1}^{K} x_i^{\alpha_i - 1}$$

où C est une constante. Ainsi, dans le cas où $\alpha = 1$, on retrouve la loi uniforme.

A priori de Jeffreys : Dans le cas binaire où le modèle de Bernoulli était utilisé, l'a priori de Jeffreys correspondait à la loi Beta(0.5,0.5). Or la loi multinomiale et la loi de Dirichlet représentent respectivement la généralisation de la loi de Bernoulli et de la loi Beta au cas multiclasses. Ainsi, l'a priori de Jeffreys pour les paramètres β_{kc}^t correspond à une loi de Dirichlet de paramètre 0.5 :

$$\beta_{kc}^t \sim Dir(0.5)$$

– **A priori sur les poids w :**
On assume toujours un a priori Gaussien avec pour moyenne 0 et pour variance 1 : $w_k \sim N(w_k|0, 1)$.

Algorithme Ignore

Le détail des étapes E et M peut être vu à la section 4.1.5. Dans ce contexte, on obtient :

E-Step : Estimation de $\mu_i^k = Pr\left[z_i = k|y_i^1, ..., y_i^T, x_i, h_i^1, ..., h_i^T, \Theta\right]$. Avec le théorème de Bayes et les équations (4.35) and (4.36), on a :

$$\mu_i^k \ \alpha \ P(z_i = k|x_i, \Theta_{z_k}) \prod_{t=1}^{T} Pr(y_i^t|z_i = k, h_i^t, \Theta_{y_k}^t) \tag{4.41}$$

$$= \frac{p_i^k \times a_i^k}{\sum_{k=0}^{K} p_i^k \ a_i^k} \tag{4.42}$$

M-Step : Estimation des paramètres Θ en maximisant l'espérance du logarithme de la vraisemblance suivant chaque paramètre :

$$E\left[ln \ Pr\left[Y, Z, X|H, \Theta\right]\right] \qquad \propto \qquad \sum_{i=1}^{N} \sum_{k=0}^{K} \mu_i^k ln\left[p_i^k a_i^k\right] \quad + \quad ln[Pr(\Theta)] \tag{4.43}$$

Les paramètres Θ sont estimés en calculant, par rapport à chaque paramètre, les gradients de (4.43) à zéro. Dans notre cas, si $\alpha_k^t \sim Dirichlet(\gamma_{k0}^t, \gamma_{k1}^t, ..., \gamma_{kK}^t)$ et $\beta_k^t \sim Dirichlet(\phi_{k0}^t, \phi_{k1}^t, ..., \phi_{kK}^t)$, on obtient :

$$\alpha_{kc}^t \ = \ \frac{\gamma_{kc}^t - 1 + \sum_{i \in \left\{i|h_i^t = 0\right\}} \mu_i^k \delta(y_i^t, c)}{\sum_{c=0}^{K} \gamma_{kc}^t - (K+1) + \sum_{i \in \left\{i|h_i^t = 0\right\}} \mu_i^k} \tag{4.44}$$

$$\beta_{kc}^t \;=\; \frac{\phi_{kc}^t - 1 + \sum_{i \in \{i|h_i^t=1\}} \mu_i^k \delta(y_i^t, c)}{\sum_{c=0}^K \phi_{kc}^t - (K+1) + \sum_{i \in \{i|h_i^t=1\}} \mu_i^k} \tag{4.45}$$

Concernant w_k, $\forall k \in \{0, 1, .., K\}$, on utilise comme pour le cas binaire la méthode de Newton-Raphson :

$$w_k^{q+1} = w_k^q - \eta \Omega_k^{-1} g_k \tag{4.46}$$

où g_k est le vecteur gradient, Ω_k la matrice Hessienne et η le pas. Le vecteur gradient est donné par :

$$g_k(w_k) = \sum_{i=1}^{N} \left[\mu_i^k - p_i^k\right] x_i - \Gamma_k w_k \tag{4.47}$$

et la matrice Hessienne est donnée par :

$$\Omega_k(w_k) = -\sum_{i=1}^{N} \left[p_i^k\right] \left[1 - p_i^k\right] x_i x_i^T - \Gamma_k \tag{4.48}$$

Les étapes E et M sont itérées jusqu'à convergence. On initialise μ_i^k à $\frac{1}{T}\sum_{t=1}^{T} \delta(y_i^t, k)$. On résume la méthode par l'algorithme 6.

Algorithm 6 Ignore Multiclasses avec Incertitude Totale.

1: données d'entrées : X, Y, Ω, seuil ϵ, iteration $q = 0$.
2: $\forall$k $\in \{0, 1, ..., K\}$:
3: initialisation : $\mu_k, w_k^q, \alpha_k^t, \beta_k^t$.
4: calculer : g_k, Ω_k, w_k^{q+1}.
5: **while** $\sum_{k=0}^{K} ||w_k^{q+1} - w_k^q||^2 \;\leq\; \epsilon$ **do**
6: q = q+1.
7: E-Step : estimer μ_k à l'aide de l'équation (4.41).
8: M-Step : ré-estimer $\alpha_k^t, \beta_k^t, w_k^{q+1}$ maximisant $E\left[ln\ Pr\left[Y, Z, X|H, \Theta\right]\right]$ à l'aide des équations (4.44), (4.45) et (4.46).
9: **end while**
10: données en sorties : $(\alpha_k^t, \beta_k^t, w_k)$.

Une fois les paramètres estimés dans l'algorithme EM, une nouvelle instance x_i est classée en calculant $p(z_i = k|x_i) = (1 + exp(-w_k^T x_i))^{-1}$, $\forall k \in \{0, 1, ..., K\}$, la probabilité que x_i ait le label k. La classe avec la probabilité la plus élevée correspond alors au réel label de x_i estimé par le classifieur.

Nous allons étudier à présent la dernière extension du modèle. Ignore multiclasses avec incertitude partielle correspond en réalité au cadre le plus général de cette étude et à une combinaison naturelle des deux extensions précédemment développées.

4.2.3 Cas Multiclasses avec Incertitude Partielle

Formalisation du Problème et Notations

Dans ce contexte d'étude, nous avons les annotations des juges $y_i^t \in \mathcal{Y} = \{(k,u)\}_{k\in\{0,K\}}^{u\in[0,1]}$, et le réel label z_i appartient à $\mathcal{Z} = \{0,1,...,K\}$. La matrice d'incertitude H est quant à elle définie comme à l'équation 4.25. Les étapes d'estimation des réels labels Z dans ce cadre d'étude sont très similaires à l'extension Ignore multiclasses avec incertitude totale. Les quelques différences sont présentées dans la section qui suit.

Modélisation du Problème

La modélisation du problème reste la même que pour les précédents modèles Ignore, à savoir, estimer la probabilité $P(Z|X,Y,H)$.

Estimateur de Maximum a Posteriori

La probabilité $P(X,Y,Z|H,\Theta)$ est décomposée suivant l'équation (4.33) de la Section 4.2.2. La variable z suit toujours une régression logistique multinomiale de paramètres $\Theta_{z_k} = \{w_k\}\,,\ \forall k \in \{0,1,...,K\}$ (cf. équation 4.34), alors que y suit un mélange de deux distributions multinomiales de paramètres $\Theta_{y_k}^t = \{\alpha_k^t, \beta_k^t\}$. La distribution de y peut être vue à la définition 7, et finalement, la vraisemblance a posteriori à maximiser est définie à l'équation 4.38.
La section qui suit définie les a prioris choisis sur chacun des paramètres.

Distribution a Priori

$\{\alpha_k^t, \beta_k^t\}$ représentant toujours respectivement les paramètres dans les situations de certitude et de doute, les a prioris choisis sur l'ensemble des paramètres restent les mêmes que pour le modèle Ignore multiclasses binaire. Ainsi, on renvoie le lecteur à la Section 4.2.2 pour plus de détails.

Algorithme

La maximisation de la vraisemblance a posteriori définie à l'équation 4.38 se fait toujours à l'aide de l'algorithme EM. A l'étape E, on estimer la probabilité $\mu_i^k = Pr\left[z_i = k|y_i^1,...,y_i^T,x_i,h_i^1,...,h_i^T,\Theta\right]$ à l'aide de l'équation 4.41, et à l'étape M, on estime les paramètres Θ en maximisant l'espérance du logarithme de la vraisemblance suivant chaque paramètre, cf. équation (4.43).
La différence réside seulement dans le calcul des gradients pour les deux paramètres $\{\alpha_k^t, \beta_k^t\}$, puisque contrairement au cas d'ignorance totale, ici les deux paramètres sont toujours pris en compte, que l'annotateur soit dans une situation d'incertitude ou de connaissance. On obtient ainsi :

$$\alpha_{kc}^t = \frac{\gamma_{kc}^t - 1 + \sum_i (1 - h_i^t)\mu_i^k \delta(y_i^t, c)}{\sum_{c=0}^K \gamma_{kc}^t - (K+1) + \sum_i (1 - h_i^t)\mu_i^k} \quad (4.49)$$

$$\beta_{kc}^t = \frac{\phi_{kc}^t - 1 + \sum_i h_i^t \mu_i^k \delta(y_i^t, c)}{\sum_{c=0}^K \phi_{kc}^t - (K+1) + \sum_i h_i^t \mu_i^k} \quad (4.50)$$

où $\delta(u, v) = 1$ si u = v, 0 sinon. Concernant les paramètres w_k, $\forall k \in \{0, 1, .., K\}$, les équations pour son estimation sont identiques à Ignore multiclasses binaire, se référer aux équations (4.46), (4.47) et (4.48).

A présent que les 4 versions d'Ignore ont été présentées, nous allons dans la prochaine section expérimenter ce modèle sur des jeux de données synthétiques afin de comparer leur performance à de précédentes approches développées.

4.3 Expérimentations

On montre dans cette section l'importance d'intégrer l'incertitude des juges dans un contexte d'apprentissage supervisé en présence de multiples annotateurs non experts. Pour cela, on compare sur de multiples jeux de données la performance des modèles Ignore proposés avec d'autres modèles plus basiques, ne prenant pas en compte l'incertitude des juges. Dans un cadre de classification binaire, la performance de Ignore est comparée à la performance du modèle Baseline de [Raykar et al., 2010], dont le détail peut être vu à la section 2.2.3, mais aussi au modèle plus classique de régression linéaire simple, dont un rapide rappel peut être lu à l'Annexe C. Dans un contexte multiclasses, le modèle baseline de Raykar reste valable, et la régression linéaire est remplacée par la régression linéaire multinomiale. Tous ces algorithmes ont été implémentés à l'aide du logiciel R-Cran, téléchargable sur Internet [2]. Les simulations ont été effectuées sur de nombreux jeux de données valables en ligne sur le site de l'UCI Machine Learning Repository [Asuncion and Newman, 2007], où les labels réels Z sont disponibles. Le détail de chaque jeu de données peut être vu à la Table 4.1. Or ces derniers n'ont pas été annotés par de multiples juges. On doit alors effectuer au préalable une étape de simulation de la matrice d'annotations Y des juges ainsi que de la matrice H d'incertitude. Le protocole expérimental pour leur simulation dans le cadre de la classification binaire et multiclasses est détaillé dans la section qui suit.

4.3.1 Protocoles expérimentaux

Cadre de l'Incertitude Totale

Dans le cadre de l'incertitude totale, chaque jeu de données D est simulé de la façon qui suit :

2. http ://cran.r-project.org/

TABLE 4.1 – Description des Jeux de Données

Nom	Id	# Entrainement	# Test	# Variables	# Classes
Breast Tissue	1	85	21	10	6
Cardiotography	2	1701	425	23	3
Ecoli	3	269	67	8	8
Iris	4	120	30	4	3
Seeds	5	168	42	19	7
Image Segmentation	6	1848	462	19	7
Vertebral Column	7	248	62	6	3
Wine	8	143	35	13	3
Yeast	9	9	1188	496	8
Dermatology	10	293	73	34	6
Satimage	11	5148	1287	36	6
Vehicle	12	757	189	18	4
Cleveland	13	237	60	13	2
Ionosphère	14	280	71	34	2
Musk	15	380	96	167	2
Glass	16	171	43	10	2
Bupa	17	276	69	7	2
Haberman	18	244	62	3	2
Vertebral	19	248	62	6	2
Spec Heart	20	213	54	22	2
Australian	21	552	138	14	2
Housing	22	404	102	14	2
Galaxy Dim	23	3353	839	14	2

1. D est divisé en deux : un jeu d'entrainement D_{train} pour la génération du modèle et un jeu test D_{test} pour tester le modèle généré, représentant respectivement 80% et 20% du jeu de données initial.
2. D_{train} est divisé en T ensembles $\{d_1, d_2, ..., d_T\}$ à l'aide de K-means.
3. On suppose que l'annotateur t est un expert pour le jeu de données d_t, i.e :

$$\begin{cases} Erreur(t, d_t) = 0\% \\ Incertitude(t, d_t) = 0\% \end{cases}$$

   En d'autres termes, l'annotateur t donne toujours la bonne réponse et est toujours confiant pour les instances du jeu de données d_t.
4. Sur le reste du jeu de données $\bar{d_t}$, on suppose que l'annotateur t fait 10% d'erreurs et U% d'incertitude, $U \in \{0\%, 10\%, ..., 80\%, 90\%\}$:

$$\begin{cases} Erreur(t, \bar{d_t}) = 10\% \\ Incertitude(t, \bar{d_t}) = U\% \end{cases}$$

A partir de là, la matrice Y d'annotations des juges est simulée de la façon suivante : Soit $\mathcal{E}$ l'ensemble des instances avec erreurs, $\mathcal{U}$ l'ensemble des instances avec incertitude totale, et $\mathcal{R}$ les instances restantes. On a :

$$\begin{cases} Y[\mathcal{E}_i, t] = random(\{0, ..., K\} - z_i) \\ Y[\mathcal{U}_i, t] = random(0, ..., K) \\ Y[\mathcal{R}_i, t] = z_i \end{cases}$$

Concernant la matrice H d'incertitude, elle est générée comme suit :

$$\begin{cases} H[\mathcal{E}_i, t] = H[\mathcal{R}_i, t] = 0 \\ H[\mathcal{U}_i, t] = 1 \end{cases}$$

Dans les expériences, les labels incertains '?' sont remplacés en simulant aléatoirement le label. Ce procédé de simulation est justifié par le fait que l'on considère ici que l'incertitude est totale, et donc qu'elle est équivalente à de l'ignorance.

Cadre de l'Incertitude Partielle

Dans un contexte d'incertitude partielle, chaque jeu de données D est simulé de la façon qui suit :

- D est divisé aléatoirement en deux : un jeu d'entrainement D_{train} représentant 80% de D, et un jeu test D_{test} (20% de D).
- On simule T=100 annotateurs, et on divise les T annotateurs en trois groupes : $T = \{T_{exp}, T_{spam}, T_{con}\}$, où T_{exp} sont les annotateurs experts, T_{spam} sont les annotateurs spammers, et T_{cons} les annotateurs conscients, représentant respectivement (10% de T, 10% de T et 80% de T). On appelle expert un annotateur ne faisant aucune erreur en annotant et étant toujours sûr de lui. On appelle spammer un annotateur étiquetant aléatoirement toutes les instances du jeu de données et donnant un taux d'incertitude aléatoire. Enfin, un annotateur conscient est un annotateur qui est conscient de son domaine d'ignorance et de connaissance.
  Soit $\mathcal{E}$ l'ensemble des instances avec erreurs, $\mathcal{U}$ l'ensemble des instances avec incertitude partielle et $\mathcal{R}$ les instances restantes. Le pourcentage d'incertitude et d'erreur sur les labels pour chaque groupe d'annotateurs sont simulés de la façon suivante :
  - $\forall t \in T_{exp}$:
  $$\begin{cases} Erreur(t, D_{train}) = random([0, 10]) \\ Incertitude(t, D_{train}) = random([0, 50]) \end{cases}$$
  - $\forall t \in T_{cons}$:
  $$\begin{cases} Erreur(t, D_{train}) = random([0, 10]) \\ Incertitude(t, D_{train}) = U \end{cases}$$
  avec $U \in \{0, 10, ..., 90\}$.

- Pour T_{spam}, tous les labels et tous les taux d'incertitude sont simulés aléatoirement.

Ainsi, la matrice Y d'annotations des juges est générée comme suit :

- $\forall t \in T_{exp}$:

$$\begin{cases} Y[\mathcal{E}_i, t] = random(\{0, ..., K\} - z_i) \\ Y[\mathcal{U}_i, t] = z_i \\ Y[\mathcal{R}_i, t] = z_i \end{cases}$$

- $\forall t \in T_{cons}$:

$$\begin{cases} Y[\mathcal{E}_i, t] = random(\{0, ..., K\} - z_i) \\ Y[\mathcal{U}_i, t] = random(\{0, ..., K\}) \\ Y[\mathcal{R}_i, t] = z_i \end{cases}$$

- $\forall t \in T_{spam}$, $Y[i, t] = random(\{0, ..., K\})$.

Concernant la matrice H, on a :

- $\forall t \in T_{exp}$:

$$\begin{cases} H[\mathcal{E}_i, t] = random([0, 0.5[) \\ H[\mathcal{U}_i, t] = random([0.5, 1]) \\ H[\mathcal{R}_i, t] = random([0, 0.5[) \end{cases}$$

- $\forall t \in T_{cons}$:

$$\begin{cases} H[\mathcal{E}_i, t] = random([0, 0.5[) \\ H[\mathcal{U}_i, t] = random([0.5, 1]) \\ H[\mathcal{R}_i, t] = random([0, 0.5[) \end{cases}$$

- $\forall t \in T_{spam}$, $H[i, t] = random([0, 1])$.

On fait varier le pourcentage d'incertitude des annotateurs conscients de 0 à 90%, afin d'étudier son impact sur la robustesse des modèles.

4.3.2 Critères d'évaluation

Afin d'évaluer les méthodes développées dans ce travail, deux critères vont être utilisés : l'AUC (ou M-AUC, une généralisation de l'AUC dans le cas multiclasses [Hand and Till, 2001]) et le taux d'erreur de classification du modèle, pour chaque taux d'incertitude (cf. Annexe D). Afin de comparer et de tester les méthodes face à l'incertitude, on génère chacun des modèles pour chaque niveau d'incertitude U allant de 0 à 90%. Chaque modèle a été simulé 100 fois à l'aide de la méthode du bootstrap, méthode consistant à rééchantillonner aléatoirement N instances avec remise parmi les N instances présentes dans le jeu initial. Cette méthode a été initialement proposée par [Efron, 1979]. Le lecteur peut se référer à de nombreux autres ouvrages pour davantage de détails [Efron and Tibshirani, 1993, Davison and Hinkley, 1997].

4.3.3 Résultats et Analyses

On représente l'évolution de l'AUC (ou du M-AUC dans le cas multiclasses) en fonction du taux d'incertitude des annotateurs. Les résultats obtenus sur tous les jeux de données étant sensiblement les mêmes, nous présentons ci-dessous les graphiques pour deux jeux de données pour chaque contexte Ignore. Le lecteur peut se référer à l'Annexe E pour visualiser les résultats sur les autres données de l'UCI. Les résultats ont été classés suivant les 4 contextes :

- Classification binaire et incertitude totale : cf. Figure 4.2 et Tableau 4.2,
- Classification binaire et incertitude partielle : cf. Figure 4.3 et Tableau 4.3,
- Classification multimodale et incertitude totale : cf. Figure 4.4 et Tableau 4.4,
- Classification multimodale et incertitude partielle : cf. Figure 4.5 et Tableau 4.5.

Pour les 4 cadres de classification, le modèle Ignore proposé est beaucoup plus stable face à l'incertitude que le modèle de Raykar et la régression et ce, pour tous les jeux de données. La régression linéaire est, sans surprise, le modèle le moins performant. Ce dernier est très sensible face à l'incertitude, et la qualité du modèle se dégrade très rapidement. Ceci peut être très bien remarqué sur plusieurs jeux de données, tels que Galaxy Dim ou Cleveland sur la Figure 4.2 dans un contexte binaire avec incertitude totale, ou encore sur les Figures 4.5 pour les jeux de données Iris et Seeds dans un contexte de classification multiclasses avec incertitude partielle. Il est d'ailleurs à remarquer que sur ces 2 dernières figures l'AUC du modèle baseline de Raykar ainsi que l'AUC de la régression s'écroulent totalement comparé à nos modèles Ignore : par exemple pour Iris, l'AUC passe de 0.95 à 0.73 pour Raykar, et de 0.95 à 0.75 pour la régression, contre un passage de 0.95 à 0.85 environ pour Ignore. Ainsi, malgré le fait que sur la plupart des jeux de données le modèle Baseline reste tout de même meilleur que le modèle de régression, il reste néanmoins plus fragile face à un taux d'incertitude élevé comparé à Ignore ; Au delà de 60% d'incertitude, le modèle de Raykar s'effondre. Ceci peut être vu sur toutes les figures, comme par exemple sur le jeu de donnée Bupa 4.3, où à partir d'un taux d'incertitude supérieur à 0.6, l'AUC de la méthode de Raykar passe de 0.78 à 0.61.
On remarque enfin que tous les modèles Ignore sont de performance équivalente, les trois a priori choisis étant aussi performants les uns que les autres. Ces résultats vont être aussi confirmés à l'aide de l'estimation du taux d'erreur de classification.

Nous testons chaque classifieur sur les jeux tests et nous calculons le taux d'erreur de classification moyen obtenu pour chaque taux d'incertitude $U \in \{0, ..., 0.9\}$. Dans le but d'avoir une vision claire des résultats obtenus, nous calculons en moyenne le taux d'erreur obtenu à travers tous les taux d'incertitude, pour chaque jeu de données. Nous obtenons ainsi les résultats présentés dans les tableaux 4.2, 4.3, 4.4 et 4.5. Les résultats obtenus confirment les résultats obtenus lors de l'étude de l'évolution de l'AUC, à savoir que : la régression linéaire est la méthode la moins efficace pour ce type de données, puisque en moyenne pour les 4 contextes, elle fait plus de 34% d'erreurs de classification, contre un pourcentage d'erreurs de classification autour de 0.27 pour Raykar, et un taux

d'erreur de classification autour de 0.22 pour les trois modèles Ignore.
Ainsi, d'un point de vue classification, les trois modèles Ignore développés dans ce travail se retrouvent être de meilleurs classifieurs, dans un contexte de classification en présence de multiples annotateurs non experts, de performance hétérogènes. On a ainsi montré à travers ces différentes expérimentations l'importance de l'intégration de l'incertitude dans un contexte d'apprentissage en présence d'annotateurs hétérogènes.

4.4 Conclusion

Le modèle Ignore développé dans ce chapitre répond au problème de la classification en présence de multiples annotateurs experts ou non. Contrairement aux méthodes précédemment développées dans ce contexte, l'originalité de Ignore est qu'il donne la possibilité aux juges d'exprimer leur incertitude lors de l'étape d'annotations. Ainsi, Ignore représente une approche probabiliste bayésienne qui intègre l'incertitude des annotateurs lors de la génération du classifieur. Ce dernier estime finalement la performance de chaque annotateur conditionnellement à leur incertitude, et prédit le réel label pour une nouvelle instance donnée.
De plus, l'avantage de l'approche Ignore proposée est qu'elle répond à quatre contextes de classification différents : la classification binaire ou multiclasses, et la classification en présence d'incertitude totale ou partielle, toutes ces extensions se faisant très naturellement et simplement. Les différentes expérimentations sur de multiples jeux de données ont montré l'importance de l'intégration de l'incertitude lors de la génération du classifieur, puisque Ignore est nettement meilleur que d'autres approches baseline, que ce soit en terme de stabilité ou de performance, surtout dans un contexte où de multiples annotateurs sont non experts.
Le chapitre qui suit présente la 2ème contribution de cette thèse, à savoir le modèle X-Ignore, où non seulement l'incertitude des annotateurs est prise en compte dans la génération du classifieur, mais aussi la qualité des données.

Ignore Binaire et Incertitude Totale

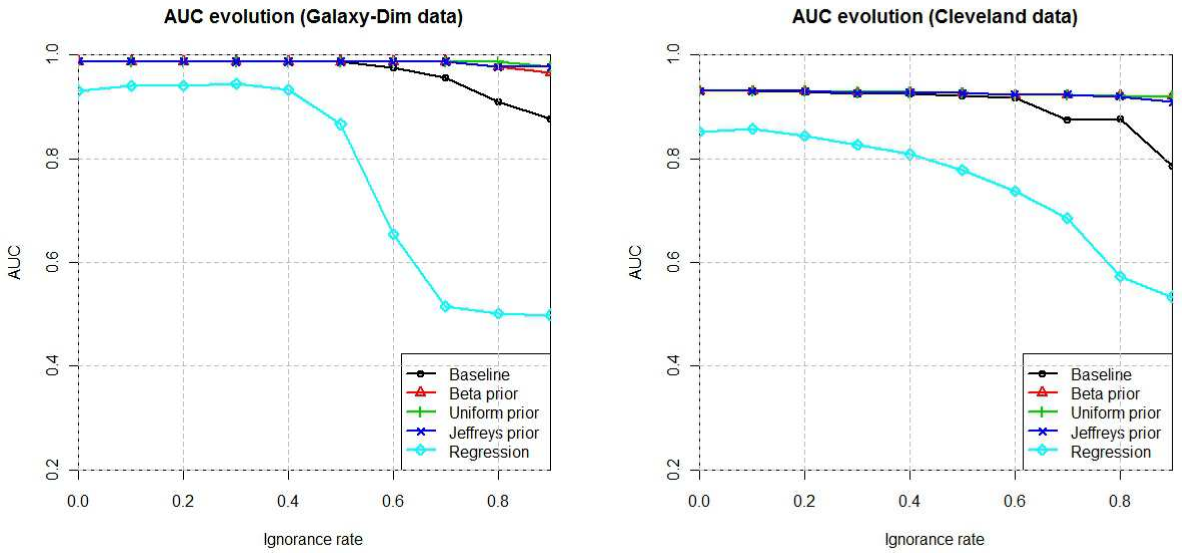


FIGURE 4.2 – Ignore Binaire et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

TABLE 4.2 – Ignore Binaire et Incertitude Totale : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	error rate cross different levels of uncertainty				
Id	Baseline	Beta	Uniform	Jeffreys	Regression
13	0.164 ± 0.038	0.135 ± 0.019	0.137 ± 0.016	0.145 ± 0.016	0.313 ± 0.096
14	0.271 ± 0.088	0.215 ± 0.012	0.218 ± 0.011	0.224 ± 0.009	0.227 ± 0.044
15	0.331 ± 0.041	0.161 ± 0.033	0.161 ± 0.032	0.162 ± 0.023	0.234 ± 0.035
16	0.391 ± 0.066	0.358 ± 0.016	0.361 ± 0.017	0.359 ± 0.022	0.385 ± 0.062
17	0.371 ± 0.025	0.361 ± 0.012	0.366 ± 0.016	0.352 ± 0.011	0.379 ± 0.023
18	0.411 ± 0.027	0.389 ± 0.021	0.379 ± 0.024	0.391 ± 0.018	0.412 ± 0.168
19	0.429 ± 0.094	0.246 ± 0.015	0.234 ± 0.016	0.239 ± 0.022	0.478 ± 0.071
20	0.345 ± 0.097	0.294 ± 0.015	0.301 ± 0.019	0.298 ± 0.024	0.390 ± 0.082
23	0.336 ± 0.083	0.063 ± 0.001	0.058 ± 10^{-5}	0.060 ± 0.001	0.625 ± 0.081
Mean	**0.339 ± 0.066**	**0.247 ± 0.016**	**0.246 ± 0.017**	**0.248 ± 0.018**	**0.383 ± 0.074**

Ignore Binaire et Incertitude Partielle

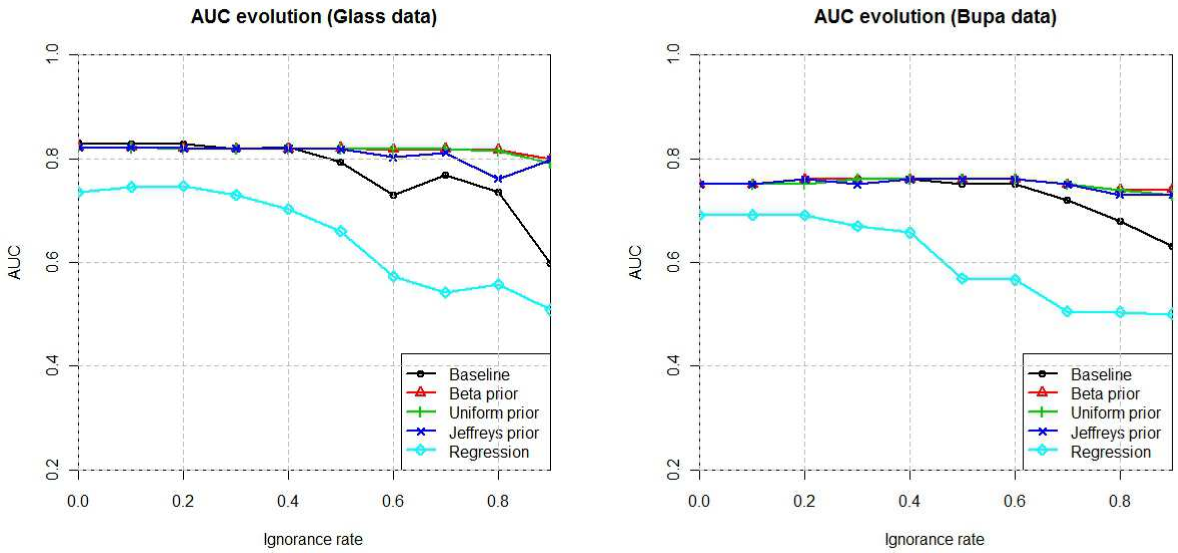


FIGURE 4.3 – Ignore Binaire et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression linéaire, en fonction du taux d'incertitude des annotateurs.

TABLE 4.3 – Ignore Binaire et Incertitude Partielle : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	error rate cross different levels of uncertainty				
Id	Baseline	Beta	Uniform	Jeffreys	Regression
13	0.214 ± 0.037	0.175 ± 0.029	0.173 ± 0.021	0.171 ± 0.024	0.321 ± 0.199
14	0.343 ± 0.076	0.222 ± 0.062	0.221 ± 0.061	0.224 ± 0.069	0.426 ± 0.098
15	0.332 ± 0.045	0.287 ± 0.039	0.280 ± 0.038	0.282 ± 0.040	0.487 ± 0.240
16	0.241 ± 0.076	0.217 ± 0.029	0.229 ± 0.028	0.214 ± 0.022	0.338 ± 0.129
17	0.134 ± 0.014	0.112 ± 0.012	0.110 ± 0.011	0.115 ± 0.011	0.187 ± 0.212
18	0.312 ± 0.032	0.298 ± 0.022	0.294 ± 0.019	0.295 ± 0.023	0.412 ± 0.168
19	0.122 ± 0.084	0.110 ± 0.065	0.121 ± 0.064	0.122 ± 0.060	0.156 ± 0.162
20	0.354 ± 0.087	0.321 ± 0.025	0.316 ± 0.021	0.318 ± 0.022	0.402 ± 0.210
23	0.345 ± 0.083	0.243 ± 0.075	0.241 ± 0.066	0.239 ± 0.070	0.450 ± 0.118
Mean	**0,266 ± 0,059**	**0,221 ± 0,039**	**0,220 ± 0,036**	**0,220 ± 0,037**	**0,353 ± 0,171**

Ignore Multiclasses et Incertitude Totale

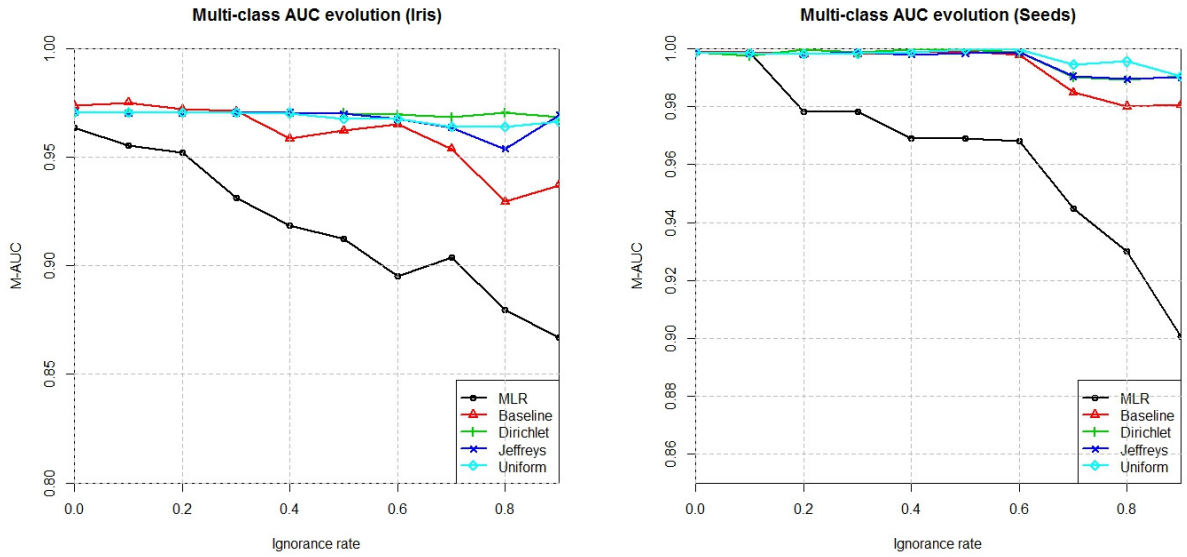


FIGURE 4.4 – Ignore Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

TABLE 4.4 – Ignore Multiclasses et Incertitude Totale : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	Error rate across different levels of uncertainty				
Id	MLR	Baseline	Dirichlet	Jeffreys	Uniform
1	0.622 ± 0.006	0.561 ± 0.003	0.419 ± 0.004	0.414 ± 0.004	0.404 ± 0.003
2	0.567 ± 0.005	0.541 ± 0.004	0.495 ± 0.001	0.493 ± 0.001	0.493 ± 0.001
3	0.247 ± 0.013	0.210 ± 0.008	0.163 ± 10^{-4}	0.166 ± 10^{-4}	0.171 ± 10^{-4}
4	0.228 ± 0.002	0.192 ± 0.001	0.174 ± 10^{-4}	0.176 ± 10^{-4}	0.173 ± 10^{-4}
5	0.119 ± 0.003	0.071 ± 0.001	0.066 ± 10^{-4}	0.067 ± 10^{-4}	0.057 ± 10^{-4}
6	0.354 ± 0.003	0.298 ± 10^{-4}	0.096 ± 10^{-4}	0.100 ± 10^{-4}	0.107 ± 0.001
7	0.613 ± 0.003	0.587 ± 0.003	0.530 ± 10^{-4}	0.539 ± 10^{-4}	0.535 ± 10^{-4}
8	0.247 ± 0.013	0.210 ± 0.008	0.166 ± 10^{-4}	0.166 ± 10^{-4}	0.171 ± 10^{-4}
9	0.539 ± 0.005	0.461 ± 0.002	0.448 ± 10^{-4}	0.448 ± 10^{-4}	0.449 ± 10^{-4}
10	0.216 ± 0.020	0.187 ± 0.014	0.081 ± 10^{-4}	0.081 ± 10^{-4}	0.082 ± 10^{-4}
11	0.236 ± 0.008	0.180 ± 0.001	0.156 ± 10^{-4}	0.145 ± 10^{-4}	0.172 ± 10^{-4}
12	0.294 ± 0.005	0.268 ± 0.002	0.222 ± 10^{-4}	0.232 ± 10^{-4}	0.225 ± 10^{-4}
Mean	$\mathbf{0.356 \pm 0.007}$	$\mathbf{0.314 \pm 0.004}$	$\mathbf{0.252 \pm 10^{-4}}$	$\mathbf{0.252 \pm 10^{-4}}$	$\mathbf{0.253 \pm 10^{-4}}$

Ignore Multiclasses et Incertitude Partielle

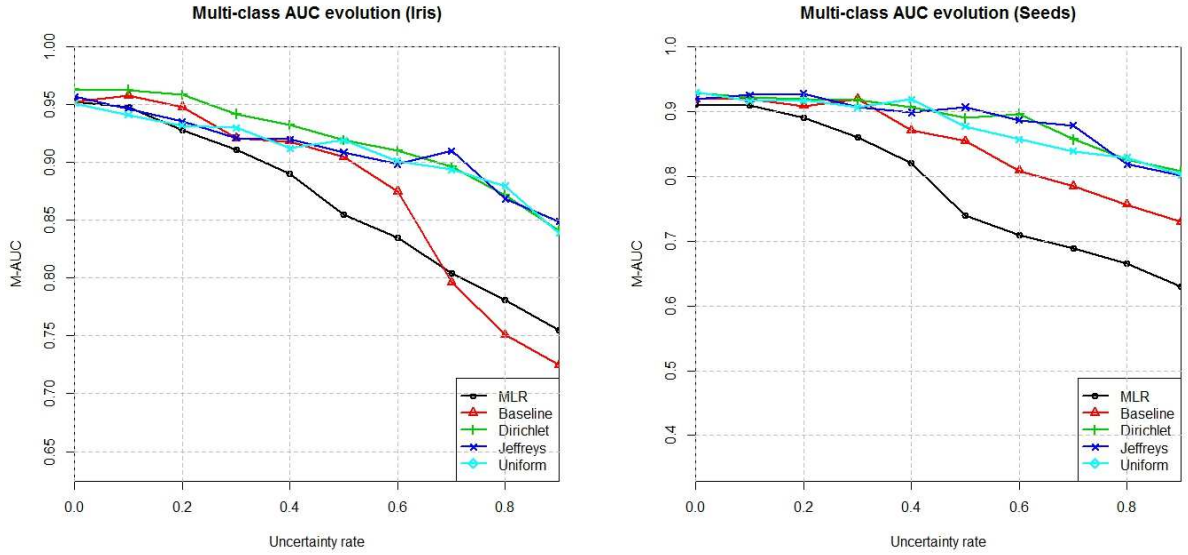


FIGURE 4.5 – Ignore Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

TABLE 4.5 – Ignore Multiclasses et Incertitude Partielle : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	Error rate across different levels of uncertainty				
Id	MLR	Baseline	Dirichlet	Jeffreys	Uniform
1	0.247 ± 0.016	0.217 ± 0.005	0.189 ± 0.004	0.184 ± 0.004	0.182 ± 0.003
2	0.248 ± 0.009	0.207 ± 0.004	0.182 ± 0.002	0.179 ± 0.002	0.175 ± 0.002
3	0.395 ± 0.009	0.322 ± 0.006	0.273 ± 0.002	0.272 ± 0.002	0.277 ± 0.001
4	0.373 ± 0.017	0.296 ± 0.007	0.252 ± 0.002	0.253 ± 0.002	0.243 ± 0.003
5	0.221 ± 0.010	0.198 ± 0.009	0.139 ± 0.005	0.130 ± 0.004	0.136 ± 0.005
6	0.203 ± 0.009	0.164 ± 0.004	0.125 ± 0.002	0.113 ± 0.002	0.116 ± 0.001
7	0.228 ± 0.008	0.184 ± 0.004	0.134 ± 0.001	0.145 ± 0.001	0.156 ± 0.002
8	0.203 ± 0.004	0.171 ± 0.003	0.135 ± 0.001	0.136 ± 0.001	0.133 ± 0.001
9	0.324 ± 0.005	0.259 ± 0.005	0.216 ± 0.002	0.207 ± 0.001	0.198 ± 0.001
10	0.322 ± 0.016	0.253 ± 0.009	0.223 ± 0.003	0.201 ± 0.006	0.196 ± 0.005
11	0.339 ± 0.015	0.294 ± 0.009	0.227 ± 0.004	0.223 ± 0.004	0.221 ± 0.005
12	0.310 ± 0.043	0.177 ± 0.004	0.160 ± 0.002	0.174 ± 0.002	0.167 ± 0.001
Mean	**0.284 ± 0.161**	**0.228 ± 0.005**	**0.188 ± 0.003**	**0.184 ± 0.002**	**0.183 ± 0.003**

Chapitre 5

Apprentissage à Partir d'Annotateurs Naifs et de Données Incertaines

Sommaire

Résumé : Le modèle Ignore développé dans le chapitre précédent a permis de constater l'importance de prendre en considération l'incertitude lors de la génération d'un classifieur en présence de multiples annotateurs naïfs. En restant toujours dans un contexte de classification supervisée en présence de multiples annotateurs, ce chapitre présente la seconde contribution de notre thèse, à savoir le modèle X-Ignore, modèle estimant et intégrant la qualité des données en plus de l'incertitude des annotateurs. L'intégration de ce nouveau critère lors de l'apprentissage est motivé par le fait que la performance des annotateurs ne dépend pas seulement de leur niveau de connaissance, mais aussi de la qualité des données observées. En effet, en présence

de données de qualité moyenne, un annotateur, même expert, peut être en difficulté à annoter. A l'inverse, des instances de très bonnes qualités seront bien annotées par la plupart des annotateurs, même non experts. Ainsi, le modèle X-Ignore développé dans cette partie décrit une approche probabiliste dans le cas d'une classification supervisée en présence de multiples annotateurs incertains. X-Ignore estime la performance des annotateurs et la qualité de chaque instance présente dans le jeu de données, et génère le classifieur en prenant en considération ces deux critères. Comme pour Ignore, 4 modèles X-Ignore ont été développés, suivant que l'on se trouve dans un contexte de classification binaire ou multiclasses, et suivant que l'incertitude des annotateurs soit totale ou partielle. Les multiples expérimentations effectuées sur des jeux de données de L'UCI valident la performance de X-Ignore. X-Ignore binaire avec incertitude totale a été publié dans [Wolley and Quafafou, 2012a].

5.1 X-Ignore Binaire avec Incertitude Totale

5.1.1 Formulation du Problème et Notations

Afin de faciliter la compréhension de ce rapport, on adopte pour X-Ignore les mêmes notations que pour le modèle Ignore. Soit la matrice $X = \left[x_1^T; ...; x_N^T\right] \in R^{N \times D}$ représentant le jeu de données de départ, $Y = \left[y_1^{(1)}, ..., y_1^{(T)}; ...; y_N^{(1)}, ..., y_N^{(T)}\right] \in R^{N \times T}$ la matrice composée des annotations des juges, et $H = \left[h_1^{(1)}, ..., h_1^{(T)}; ...; h_N^{(1)}, ..., h_N^{(T)}\right] \in R^{N \times T}$ leur matrice d'incertitude. On rappelle que l'on se place ici dans un cadre de classification binaire, où les annotateurs sont tenus d'ajouter le caractère ' ?' avec le label en cas d'incertitude. Ainsi, $y_i^t \in \mathcal{Y} = \{0, 1\} \cup \{(k, ?)\}_{k=0}^{1}$. Enfin, $Z = [z_1, ..., z_N]^T$ correspond au vecteur des labels réels (inconnu). On rappelle alors que dans le cas d'incertitude totale, la matrice H est définie comme suit :

$$h_i^t = \begin{cases} 0 & \text{si } y_i^t = \{0, 1\} \\ 1 & \text{sinon.} \end{cases} \tag{5.1}$$

Conditionnellement à l'incertitude des annotateurs, notre objectif est :
- D'estimer les labels réels Z,
- De produire un classifieur pour prédire le label z d'une nouvelle instance x,
- D'estimer la performance de chaque annotateur,
- D'estimer la qualité de chaque instance et d'intégrer ces estimations dans la génération du classifieur.

5.1.2 Modélisation du Problème et Notations

L'objectif est de maximiser la probabilité d'obtenir les labels réels Z, connaissant les données X, les labels des annotateurs Y ainsi que leur incertitude. Ainsi, notre but est de maximiser la distribution conditionnelle jointe $P(Z|X, Y, H)$. L'originalité de X-Ignore par rapport à Ignore est de supposer que la performance des annotateurs dépend aussi

de la qualité des instances des données observées. On construit un modèle probabiliste qui décrit les relations entre l'ensemble des variables x, y, z et h. La structure du modèle X-Ignore peut être vue à la la Figure 5.1. Une expression explicite pour $P(Z|X,Y,H)$ étant inconnue, on fait de nouveau appel à une méthode générative (cf Section 3.1). D'après la loi de probabilité jointe et la règle de Bayes, on a :

$$P(Z|X,H,Y) = \frac{P(Z,Y|X,H)}{P(Y|X,H)} \tag{5.2}$$

$$= \frac{P(Y|Z,X,H)P(Z|X,H)}{P(Y|X,H)} \tag{5.3}$$

$$\propto P(Y|Z,X,H)P(Z|X,H) \tag{5.4}$$

Or

$$P(Y|Z,X,H)P(Z|X,H) = P(Y,Z|X,H)$$

Donc

$$P(Z|X,H,Y) \propto P(Y,Z|X,H) \tag{5.5}$$

Maximiser la probabilité $P(Z|X,Y,H)$ revient donc à maximiser $P(Y,Z|X,H)$. On effectue cette maximisation en plusieurs étapes :

1. On décompose $P(Y,Z|X,H)$ suivant Z et Y, puis on choisit des distributions sur les deux variables, ce qui engendre des paramètres Θ à estimer,
2. On applique une approche bayésienne dans le but d'intégrer l'incertitude H des annotateurs : des a prioris sont fixés sur chacun des paramètres Θ suivant la confiance accordée par les annotateurs à chacun des labels,
3. Les paramètres Θ sont estimés à l'aide de l'estimateur de maximum a posteriori :

$$\hat{\Theta}_{MAP} = argmax_{\Theta} \{lnP[Y,Z|X,H,\Theta] + lnP[\Theta]\} \tag{5.6}$$

   La maximisation de $\hat{\Theta}_{MAP}$ se fait à l'aide de l'algorithme EM (Expectation-Maximisation algorithm) (cf. Section 3.4), les labels réels Z étant manquants.

Chacune de ces étapes est détaillée dans la section qui suit.

5.1.3 Estimateur de Maximum a Posteriori

On modélise le problème par la distribution conditionnelle jointe $P(Y,Z|X,H)$. A l'aide du théorème de Bayes et en supposant que les annotateurs (resp. les instances) sont indépendants entre eux (resp. indépendantes entre elles), la distribution conditionnelle jointe peut s'écrire :

$$P(Y,Z|X,H,\Theta) = \prod_{i}^{N}\prod_{t}^{T} P[y_i^t|x_i,z_i,h_i^t,\Theta_y]P(z_i=1|x_i,h_i^t,\Theta_z) \tag{5.7}$$

avec $\Theta = \{\Theta_z,\Theta_y\}$ les paramètres à estimer. Afin de définir ces paramètres, il nous faut fixer dans un premier temps une distribution pour chacune des variables y et z.

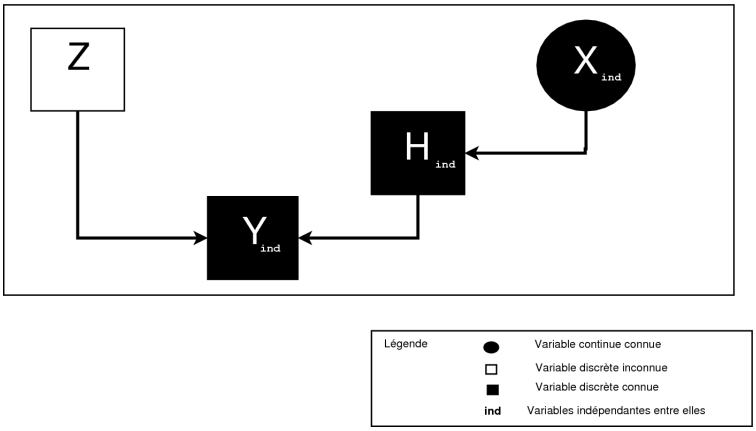


FIGURE 5.1 – Structure du Modèle X-Ignore

Concernant la variable z représentant le réel label des instances, il est très naturel de considérer que cette dernière ne dépend pas de la performance des annotateurs et de leur taux de connaissance. Ainsi, on peut écrire $Pr(z_i = 1|x_i, h_i^t, \Theta_z) = Pr(z_i = 1|x_i, \Theta_z)$. La variable z peut alors suivre n'importe quelle distribution de probabilité ; on choisit pour ce cadre binaire la régression logistique par souci de simplification. Ainsi, on a :

$$P[z_i = 1|x_i, \Theta_z] = \frac{1}{1 + e^{-w^T x_i}} \tag{5.8}$$

où $w \in R^d$. Par conséquent, $\Theta_z = \{w\}$.

Concernant la variable y, on pose un modèle Bernoulli de paramètre η, représentant la probabilité que l'annotateur t donne la bonne réponse. $\eta \in [0, 1]$, et $P[y_i^t|x_i, z_i, h_i^t, \Theta_y]$ peut s'écrire sous la forme :

$$P(y_i^t|x_i, z_i, h_i^t, \Theta_y) = (1 - \eta_i^t)^{|y_i^t - z_i|}(\eta_i^t)^{1-|y_i^t - z_i|} \tag{5.9}$$

Le modèle X-Ignore suppose que la probabilité qu'un annotateur réponde correctement dépend de trois critères : sa compétence, son taux d'incertitude, ainsi que de la qualité de chaque instance observée. Ainsi, le paramètre η doit également dépendre de ces trois critères. On définit alors $\beta_i \in [0, +\infty[$ le paramètre qui représente la qualité de l'instance i, et $\alpha^t \in]-\infty, +\infty[$ le paramètre qui représente la compétence de l'annotateur t. η dépend alors de ces deux paramètres ainsi que de l'incertitude h. On pose alors :

$$\eta_i^t = \frac{h_i^t}{1 + exp(-\alpha_1^t \beta_i)} + \frac{1 - h_i^t}{1 + exp(-\alpha_0^t \beta_i)} \tag{5.10}$$

où

$$\beta_i \longrightarrow \begin{cases} 0 & \text{si } x_i \text{ est très difficile à étiqueter,} \\ +\infty & \text{si } x_i \text{ est très facile à étiqueter.} \end{cases} \tag{5.11}$$

Dans les situations certaines, on a :

$$\alpha_0^t \longrightarrow \begin{cases} -\infty & \text{si l'annotateur t a toujours tort,} \\ +\infty & \text{si l'annotateur t a toujours raison,} \\ 0 & \text{si l'annotateur t annote aléatoirement.} \end{cases} \tag{5.12}$$

Reciproquement, dans les situations d'incertitude, on a :

$$\alpha_1^t \longrightarrow \begin{cases} -\infty & \text{si l'annotateur t a toujours tort,} \\ +\infty & \text{si l'annotateur t a toujours raison,} \\ 0 & \text{si l'annotateur t annote aléatoirement.} \end{cases} \tag{5.13}$$

Ainsi, notre modèle estime séparément la performance des annotateurs dans les situations d'incertitude totale et de connaissance. Ceci permet par la suite de connaitre la fiabilité des annotateurs, un annotateur sérieux devant logiquement avoir une performance élevée dans les situations de certitude. Ainsi, on a $\alpha = \{\alpha_0, \alpha_1\}$, où α_0 (resp. α_1) est la performance de l'annotateur t dans les situations de connaissance (resp. d'incertitude).
On remarque alors que si le paramètre α_0 (ou α_1) est égal à 0 (c'est-à-dire si l'annotateur annote aléatoirement), alors quelque soit la qualité β_i de l'instance, la probabilité η est égale à 0.5. Ce résultat reflète bien l'aléa des labels donnés. De même, plus la valeur du paramètre est elevé (c'est-à-dire plus l'annotateur est expert), plus la probabilité η qu'il réponde correctement est élevée.

Concernant β, une valeur proche de 0 signifie que l'instance est très ambigüe, et qu'elle sera alors difficile à étiqueter même dans le cas où l'annotateur est expert. A l'inverse, une très grande valeur de β signifie que l'instance est très facile à annoter, même dans le cas où le juge est non expert. On note que si la valeur de β est égale à 0, on retrouve une probabilité η égale à 0 quelque soit la performance de l'annotateur. Cela reflète bien la difficulté de l'instance à être étiquetée. A l'inverse, pour une très grande valeur du paramètre, η sera proche de 1 quelque soit la performance du juge.

Afin de s'assurer que le paramètre β soit dans l'intervalle $[0, +\infty[$, on pose :

$$\beta_i = \begin{cases} exp(\gamma_1^i) & \text{si } \frac{1}{T}\sum_{t=1}^{T} h_i^t > 0.5 \\ exp(\gamma_0^i) & \text{sinon} \end{cases} \tag{5.14}$$

avec $\gamma = \{\gamma_0, \gamma_1\} \in]-\infty, +\infty[$. En d'autre termes, on discrimine le paramètre β en fonction du nombre de juges ayant annoté l'instance avec certitude. On suppose qu'une instance est a priori difficile si plus de la moitié des juges sont dans l'incertitude quant au label à donner. A l'inverse, on suppose qu'une instance est a priori facile si plus de la moitié des annotateurs sont certains du label qu'ils ont attribué. Cette discrimination est effectuée afin de pouvoir fixer, par la suite, des distributions a priori sur les paramètres.

Finalement, on se retrouve avec $\Theta_y = \{\alpha_0, \alpha_1, \gamma_0, \gamma_1\}$, et notre objectif est d'estimer tous les paramètres $\Theta = \{\Theta_z, \Theta_y\}$. On fixe des a prioris sur les paramètres en fonction

de l'incertitude ou de la connaissance des annotateurs, et l'estimation des paramètres est alors effectuée en maximisant le logarithme de vraisemblance a posteriori à l'aide de l'estimateur du maximum a posteriori (MAP) :

$$\hat{\Theta}_{MAP} = argmax_{\Theta} \{ln\ P[Y, Z|X, H, \Theta] + ln\ P[\Theta]\} \tag{5.15}$$

La prochaine section décrit les a prioris fixés sur chacun des paramètres.

5.1.4 Distribution a Priori

Comme tous les paramètres $\{\alpha_0, \alpha_1, \gamma_0, \gamma_1, w\}$ sont dans l'intervalle $]-\infty, +\infty[$, on pose par souci de simplification un a priori Gaussien sur l'ensemble des paramètres. La moyenne μ et la variance τ^2 de l'a priori Gaussien pour chaque paramètre sont fixées de la manière suivante :

- $\{\alpha_0, \alpha_1\}$: Ces paramètres estiment la performance des annotateurs dans le cas d'incertitude et de connaissance totale. Ainsi, on suppose que dans le cas d'incertitude totale, les juges annotent aléatoirement les instances données, et il y a alors équiprobabilité entre tous les labels. On pose alors $\mu_{\{\alpha_1\}} = 0$ dans le but de refléter l'aléa des labels considérés (cf. équation (5.13)).
  A l'inverse, dans les situations sûres, on suppose que les annotateurs sont certains du label attribué. Il nous faut alors fixer une valeur plus élevée pour μ, afin de refléter la confiance accordée au label donnée (cf. équation (5.12)). Suite à différentes expérimentations, on choisit empiriquement d'attribuer la valeur $\mu_{\{\alpha_0\}} = 2$. Enfin, pour la variance, on pose pour les deux paramètres $\tau^2_{\{\alpha_0,\alpha_1\}} = 0.01$, reflétant ainsi une haute confiance sur les différents a priori.

- $\{\gamma_0, \gamma_1\}$: Ces paramètres estiment la qualité des instances du jeu de données. Comme γ_0 reflète une bonne qualité de l'instance, on fixe $\mu_{\{\gamma_0\}}$ à 2. A l'opposé, pour γ_1, $\mu_{\{\gamma_1\}} = -2$. La variance $\tau^2_{\{\gamma_0,\gamma_1\}}$ est aussi fixée à 0.01.

- w : Afin de mettre un a priori sur tous les paramètres, on suppose un a priori Gaussien sur les poids w, de moyenne zéro et de variance $\tau^2_{\{w\}} = 1$.

Le choix des valeurs attribuées aux différents paramètres a été fait empiriquement, suite à de multiples expérimentations.
Il nous faut à présent estimer les paramètres $\Theta_y = \{\alpha_0, \alpha_1, \gamma_0, \gamma_1\}$

5.1.5 Algorithme X-Ignore

Notre but est d'estimer $\hat{\Theta}_{MAP}$ défini par l'équation 5.15. En présence des valeurs manquantes z, une approche largement utilisée pour estimer le logarithme de maximum a posteriori est l'algorithme EM (Expectation Maximisation) [Dempster et al., 1977], aussi utilisé dans le modèle Ignore développé dans le chapitre précédent. On invite le lecteur à se référer à la Section 3.4 pour de plus amples informations. Ainsi, on applique

EM au cas du modèle X-Ignore, ce qui donne les deux étapes suivantes :

Etape E : On pose (par definition) $\tilde{p}(z_i) = p(z_i|x_i, y_i, h_i, \Theta)$. Selon le Theorème de Bayes, $\tilde{p}(z_i) \propto p(z_i, y_i|x_i, h_i, \Theta)$ et la distribution conditionnelle jointe peut être exprimée par :

$$\tilde{p}(z_i) \ \propto \prod_t p(z_i|x_i, \Theta_z) p(y_i^{(t)}|x_i, z_i, h_i^{(t)}, \Theta_y) \tag{5.16}$$

Estimer $\tilde{p}(z_i)$ à l'aide de l'équation ci-dessus.

Etape M : Maximiser

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

Il est difficile de ramener les gradients de cette expression suivant chaque paramètre (les calculs sont délicats). La maximisation de cette expression n'étant pas évidente, on applique l'algorithme LBFGS quasi-Newton [Nocedal and Wright, 2003] pour résoudre le problème :

$$\begin{aligned} \max_{\Theta}\ f_{opt}(\Theta) = \max_{\Theta} \sum_{i,t} E_{\tilde{p}(z_i)}[log\ p(z_i|x_i, h_i^{(t)}, \Theta_z) + \\ log\ p(y_i^{(t)}|x_i, z_i, h_i^{(t)}, \Theta_y)] + log\ [p(\Theta)] \end{aligned}$$

L'algorithme LBFGS est un membre de la famille d'optimisation quasi-Newton largement utilisée dans les problèmes d'optimisation (cf. Section 3.4.1). Cet algorithme avait aussi été utilisé pour le modèle Ignore.

Pour finir, l'approximation des paramètres $\{\alpha_0, \alpha_1, \gamma_0, \gamma_1, w\}$ nécessite la réitération des étapes E et M jusqu'à convergence. L'algorithme est résumé dans l'algorithme 7 :

Algorithm 7 Algorithme X-Ignore

1: Données initiales : X, Y, H,
2: Initialiser : w, w_{new}, $\alpha_0, \alpha_1, \gamma_0, \gamma_1$ et le seuil ϵ
3: **while** $||w - w_{new}||^2 \geq \epsilon$ **do**
4: w = w_{new}
5: Step E : Estimer $\tilde{p}(z)$ à l'aide de l'équation 5.16
6: Step M : Recalculer $\alpha_0, \alpha_1, \gamma_0, \gamma_1, w_{new}$ en maximisant $\sum_t \sum_i E_{\tilde{p}(z_i)}[log\ p(y_i^{(t)}, z_i|x_i, h_i^t, \Theta)] + log\ p[\Theta]$ à l'aide de l'algorithme LBFGS quasi-Newton.
7: **end while**
8: **return** $\alpha_0, \alpha_1, \gamma_0, \gamma_1, \{w\}$

Une fois les paramètres estimés, une nouvelle instance x est classée en calculant $P(z = 1|x) = (1 + exp(-w^T x))^{-1}$, la probabilité que x ait le véritable label z égale à 1.

Si $P(z = 1|x)$ a une valeur supérieur à 0.5, alors le véritable label z=1. Sinon, z=0.

Nous allons à présent étudier les extensions du modèle X-Ignore au cas d'incertitude partielle et au cas multiclasses, extensions qui s'avèrent nécessiter très peu de changements par rapport au cas binaire avec incertitude totale.

5.2 Extensions du Modèle X-Ignore

De la même façon que pour le modèle Ignore, nous étendons le modèle X-Ignore au cas de l'incertitude partielle, et au cas de la classification multiclasses. Ces cadres d'étude généralisent alors le modèle X-Ignore précédemment développé. On montre dans cette section que cette généralisation se fait assez simplement et de manière très naturelle.
Nous reprenons pour cela toutes les étapes du modèle Ignore précédemment développé, en effectuant les changements nécessaires.

5.2.1 X-Ignore Binaire avec Incertitude Partielle

On rappelle que le cadre de l'incertitude partielle correspond à un cadre où les annotateurs associent à chaque label un taux d'incertitude compris entre 0 et 1. Ainsi, ce cadre est une généralisation du cadre développé à la section précédente, où on supposait que les annotateurs étaient totalement ignorants ou certains du label donné. Cette généralisation est immédiate dans le cas du modèle X-Ignore.

Modélisation du Problème et Notations

Soit $X \in R^{N\times D}$ la matrice des données de départ, et $Y \in R^{N\times T}$ la matrice d'annotations des juges. Dans un contexte binaire avec incertitude partielle, chaque juge annote les instances tout en précisant un degré d'incertitude compris entre 0 et 1 (0 reflétant une certitude totale, et 1 une ignorance totale). On a alors $y_i^t \in \mathcal{Y} = \{(0, u_i^t) \cup (1, u_i^t)\}$ avec $u_i^t \in [0, 1]$. On redéfinit alors la matrice d'incertitude $H \in R^{N\times T}$ dans ce contexte. $\forall i \in \{1, ..., N\}$, $\forall\, t \in \{1, ..., T\}$, on a :

$$h_i^t = u_i^t \tag{5.17}$$

Estimateur de Maximum a Posteriori

Le maximum de vraisemblance à maximiser est inchangée (cf. équation (5.15)). On reste dans un cadre de classification binaire, on laisse donc la distribution logistique pour de la variable z et un modèle de Bernoulli pour la variable y (cf. équation (5.9),(5.8)).

Distribution a Priori

Les distributions a priori sur les paramètres ne changent pas dans un contexte d'incertitude partielle (cf Section 5.1.4).

Algorithme X-Ignore

L'algorithme est équivalent à l'algorithme Ignore binaire avec incertitude totale (cf. Section 5.1.5).

5.2.2 X-Ignore Multiclasses avec Incertitude Totale

Modélisation du Problème et Notations

Soit $X \in R^{N \times D}$ la matrice des données de départ. Contrairement au cas binaire, dans ce contexte une instance x_i peut être classée dans l'une des (K+1) classes disponibles. Ainsi, dans le cas où les annotateurs ont la possibilité d'exprimer leur incertitude totale, on a $y_i^t \in \mathcal{Y} = \{0, 1, ..., K\} \cup \{(k, ?)\}_{k \in \{0,...,K\}}$. Concernant H, dans ce contexte elle est définie ci-dessous :

$$h_i^t = \begin{cases} 1 & \text{si } y_i^t = \{(k, ?)\}_{k=0}^K \\ 0 & \text{sinon} \end{cases} \tag{5.18}$$

Estimateur de Maximum a Posteriori

Le maximum de vraisemblance à maximiser est toujours défini à l'équation (5.15), les différences résidant principalement dans les distributions choisies pour les variables z et y. En effet, pour z, une régression logistique avait été optée dans le cas binaire. Or une généralisation au cas multiclasses de la régression logistique est la régression logistique multinomiale (cf. Annexe D.3). On définit ainsi dans ce modèle la distribution de z par :

$$P(z_i = k | x_i, \Theta_{z_k}) = \begin{cases} \frac{1}{1+\sum_{k=0}^{K-1} e^{w_k^T x_i}} & \text{si } k = K \\ \frac{e^{w_k^T x_i}}{1+\sum_{k=0}^{K-1} e^{w_k^T x_i}} & \text{sinon} \end{cases} \tag{5.19}$$

On a alors $\Theta_{z_k} = \{w_k\}\,,\ \forall k \in \{0, 1, ..., K\}$. Concernant y, on pose toujours un modèle de Bernoulli de paramètre η représentant la probabilité que l'annotateur est raison. Cependant, dans un contexte multiclasses, $Pr[y_i^t | x_i, z_i, h_i^t, \Theta_y]$ s'écrit sous la forme :

$$P(y_i^t | x_i, z_i, h_i^t, \Theta_y) = (1 - \eta_i^t)^{1-\delta(y_i^t, z_i)} (\eta_i^t)^{\delta(y_i^t, z_i)} \tag{5.20}$$

où $\delta(u, v) = 1$ si u = v et 0 sinon.

Les paramètres à estimer dans ce modèle sont $\left\{\alpha_0, \alpha_1, \gamma_0, \gamma_1, \{w_k\}_{k \in \{0,...,K\}}\right\}$.

Distribution a Priori

Les a prioris ne changent pas dans ce contexte (cf Section 5.1.4).

Algorithme X-Ignore

La maximisation de $\hat{\Theta}_{MAP}$ définie à l'équation 5.15 se fait selon chaque classe k disponible. Ainsi, à l'étape E de l'algorithme EM, l'objectif est d'estimer $\tilde{p}(z_i = k)$ à l'aide de l'équation 5.16, pour $k = \{0, ..., K\}$. Puis à l'étape M, on maximise

$$\sum_t \sum_i E_{\tilde{p}(z_i=k)}[log\ p(y_i^{(t)}, z_i = k | x_i, h_i^{(t)}, \Theta)] + log\ [p(\Theta)]$$

Une fois les paramètres estimés, une nouvelle instance est classée en calculant, pour toutes les classes k, $Pr(z_i = k | x_i, \Theta_{z_k})$. L'instance appartient alors à la classe avec la plus haute probabilité.

5.2.3 X-Ignore Multiclasses avec Incertitude Partielle

Modélisation du Problème et Notations

Dans un contexte multiclasses avec incertitute partielle, on a les annotations des juges $y_i^t \in \mathcal{Y} = \{(k, u)\}_{k \in \{0,...,K\}}^{u \in [0,1]}$, et le réel label z_i appartient à $\mathcal{Z} = \{0, 1, ..., K\}$. La matrice d'incertitude H est quant à elle définie comme à l'équation (5.17).

Estimateur de Maximum a Posteriori

Le maximum de vraisemblance à maximiser est toujours défini à l'équation (5.15), avec pour distributions de y et de z les équations (5.19) et (5.20).

Distribution a Priori

Les a prioris sont toujours inchangés, et peuvent être vus à la Section 5.1.4.

Algorithme X-Ignore

L'algorithme est simulaire au modèle Ignore multiclasses avec incertitude totale (cf. Section 5.2.2).

Une fois les 4 versions du modèle X-Ignore développées, il nous faut à présent les tester et les valider sur de multiples jeux de données. Ce sera l'objectif de la section qui suit.

5.3 Expérimentations

5.3.1 Protocole Expérimental et Evaluation

Dans cette section, on compare la performance du modèle X-Ignore avec 3 autres méthodes : la méthode de Raykar et Al. [Raykar et al., 2010] décrite au paragraphe 2.2.3, la méthode de Yan et Al. [Yan et al., 2010] décrite au paragraphe 2.4.3, et enfin le

modèle plus classique de régression linéaire dans le cas binaire, ou régression multimodale dans le cas multiclasses, implémentées encore une fois à l'aide du logiciel R-Cran. Parmi toutes ces méthodes, seule la méthode de Yan et Al. prend en compte la qualité des données lors de la génération du modèle. Les simulations ont été effectuées sur des données du site de l'UCI Machine Learning Repository, dont le détail peut être vu à la table 4.1 de la section 4.3.
Nous avons repris le protocole expérimental ainsi que les critères d'évaluation (taux d'erreur de classification et AUC) du modèle Ignore développé au chapitre précédent. On invite donc le lecteur à se reporter à la section 4.3 du chapitre 4 pour de plus amples informations.

5.3.2 Résultats et Analyses

Comme pour le modèle Ignore étudié au chapitre 4, nous présentons ci-dessous les résultats de l'évolution de l'AUC (ou du M-AUC) pour seulement deux jeux de données, le reste des résultats pouvant être vus au l'Annexe F. Concernant le taux d'erreur de classification, les résultats sont toujours présentés sous la forme de tableau où l'on estime le taux d'erreur de classification moyen et la variance sur le jeu test pour chaque jeu de données, puis on calcule la moyenne générale du taux d'erreur et de la variance sur toutes les données utilisées.
Les résultats sont divisés selon les 4 contextes :

- Classification binaire et incertitude totale : cf. Figure 5.2 et Tableau 5.1,
- Classification binaire et incertitude partielle : cf. Figure 5.3 et Tableau 5.2,
- Classification multimodale et incertitude totale : cf. Figure 5.4 et Tableau 5.3,
- Classification multimodale et incertitude partielle : cf. Figure 5.5 et Tableau 5.4.

Les Graphiques 5.2, 5.3, 5.4 et 5.5 montrent l'évolution de l'AUC ou du M-AUC pour les 4 modèles Raykar, Yan, X-Ignore et la régression, lorsque le taux d'incertitude des annotateurs augmente. On remarque que pour tous les jeux de données, X-Ignore gagne en stabilité par rapport aux modèles précédemment développés. En effet, on peut prendre l'exemple du jeu de données Galaxy Dim sur le Graphique 5.3, où le M-AUC s'effondre pour les trois modèles Raykar, Yan et régression, en comparaison au modèle X-Ignore. Pour Raykar et Yan, le M-AUC passe de 0.97 à 0.88, pour la régression il passe de 0.97 à 0.75, alors que pour X-Ignore, le M-AUC est assez stable puisqu'il ne baisse que de 0.01, passant de 0.99 à environ 0.98. Cette observation peut être faite sur tous les jeux de données. Ainsi, X-Ignore dispose des mêmes qualités que Ignore, avec l'avantage de pouvoir en plus estimer la qualité des instances du jeu de données. Ces résultats sont confirmés par les tableaux estimant le taux d'erreurs de classification. En moyenne sur les 4 contextes, la régression a un pourcentage d'erreurs de 0.337, contre 0.325 pour Raykar, 0.308 pour Yan et 0.275 pour X-Ignore. De plus, la stabilité du classifieur X-Ignore en termes d'erreurs de classification est aussi meilleure par rapport aux autres modèles (variance de 0.019 pour X-Ignore, contre 0.025 pour Yan, 0.022 pour Raykar et 0.027 pour la régression).

X-Ignore Binaire et Incertitude Totale

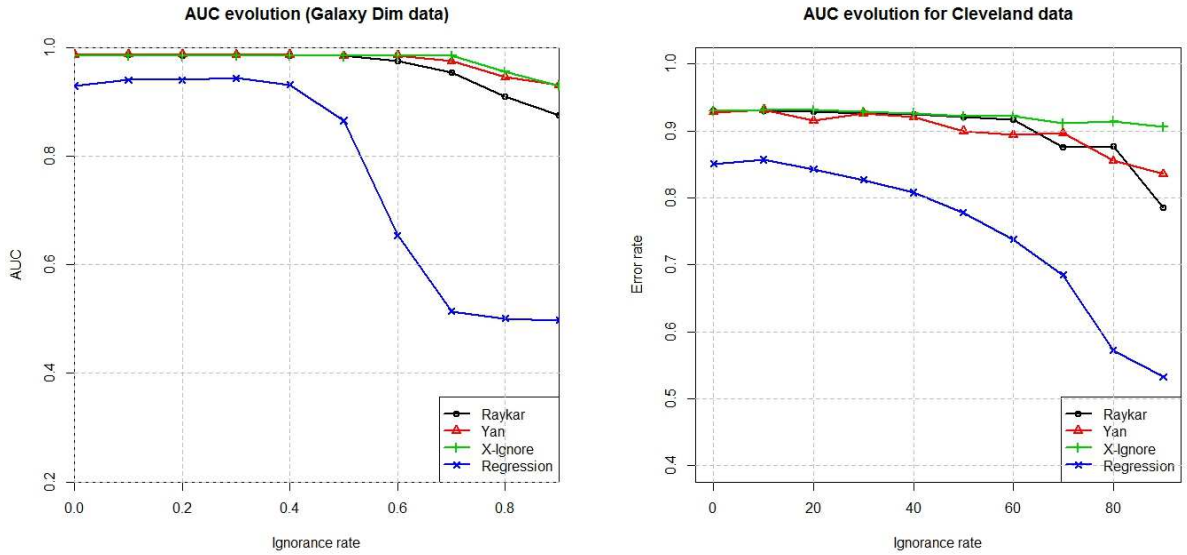


FIGURE 5.2 – Classification Binaire et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire en fonction du taux d'incertitude des annotateurs.

TABLE 5.1 – Classification Binaire et Incertitude Totale : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	Error rate cross different uncertainty levels			
Dataset	Regression	Raykar	Yan	X-Ignore
Cleveland	0.313 ± 0.096	0.164 ± 0.037	0.205 ± 0.036	0.169 ± 0.025
Galaxy Dim	0.625 ± 0.081	0.336 ± 0.083	0.322 ± 0.305	0.291 ± 0.219
Ionosphere	0.227 ± 0.044	0.270 ± 0.088	0.264 ± 0.075	0.231 ± 0.034
Musk	0.234 ± 0.035	0.331 ± 0.041	0.232 ± 0.060	0.192 ± 0.035
Glass	0.385 ± 0.062	0.390 ± 0.066	0.368 ± 0.054	0.324 ± 0.043
Bupa	0.379 ± 0.023	0.371 ± 0.025	0.362 ± 0.040	0.365 ± 0.010
Vertebral	0.478 ± 0.071	0.429 ± 0.094	0.410 ± 0.074	0.386 ± 0.085
Haberman	0.412 ± 0.168	0.411 ± 0.027	0.405 ± 0.012	0.391 ± 0.012
Mean	**0.382 ± 0.072**	**0.339 ± 0.062**	**0.322 ± 0.0778**	**0.293 ± 0.0544**

X-Ignore Binaire et Incertitude Partielle

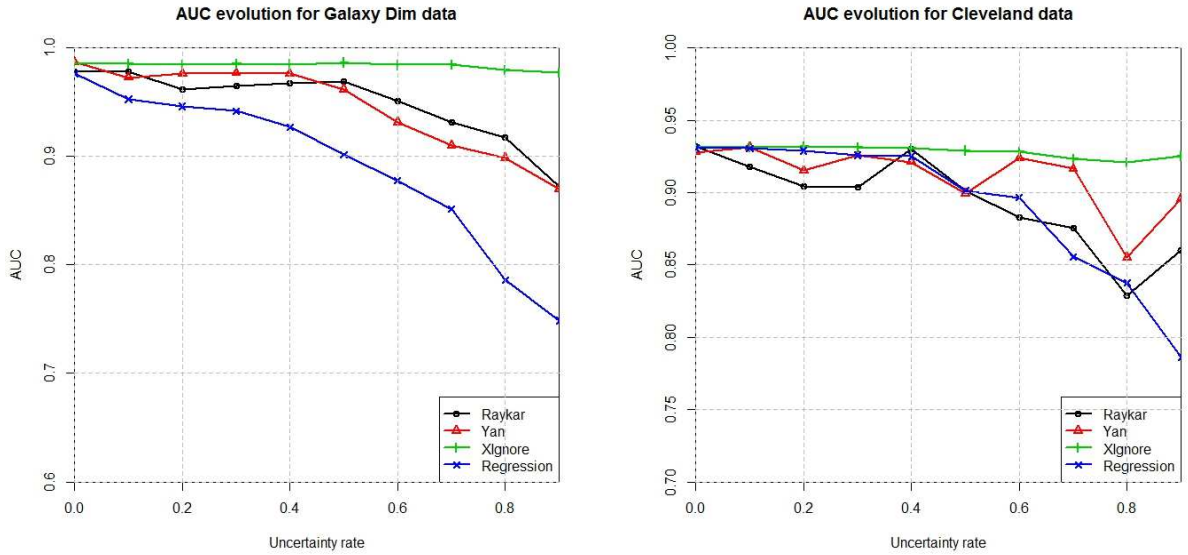


FIGURE 5.3 – Classification Binaire et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire en fonction du taux d'incertitude des annotateurs.

TABLE 5.2 – Classification Binaire et Incertitude Partielle : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	Error rate cross different uncertainty levels			
Dataset	Regression	Raykar	Yan	X-Ignore
Cleveland	0.339 ± 0.003	0.269 ± 0.001	0.263 ± 0.001	0.218 ± 0.001
Galaxy Dim	0.534 ± 0.007	0.429 ± 0.002	0.405 ± 0.003	0.360 ± 10^{-4}
Ionosphere	0.362 ± 0.004	0.314 ± 0.002	0.275 ± 0.002	0.207 ± 0.001
Musk	0.446 ± 0.031	0.409 ± 0.017	0.385 ± 0.015	0.369 ± 0.007
Glass	0.520 ± 0.007	0.406 ± 0.016	0.348 ± 0.011	0.338 ± 0.002
Bupa	0.508 ± 0.008	0.432 ± 0.002	0.430 ± 0.001	0.352 ± 10^{-4}
Vertebral	0.550 ± 0.010	0.497 ± 0.010	0.383 ± 0.015	0.310 ± 0.006
Haberman	0.479 ± 0.012	0.425 ± 0.006	0.384 ± 0.009	0.377 ± 0.002
Mean	$\mathbf{0.467 \pm 0.010}$	$\mathbf{0.397 \pm 0.007}$	$\mathbf{0.359 \pm 0.007}$	$\mathbf{0.316 \pm 0.002}$

X-Ignore Multiclasses et Incertitude Totale

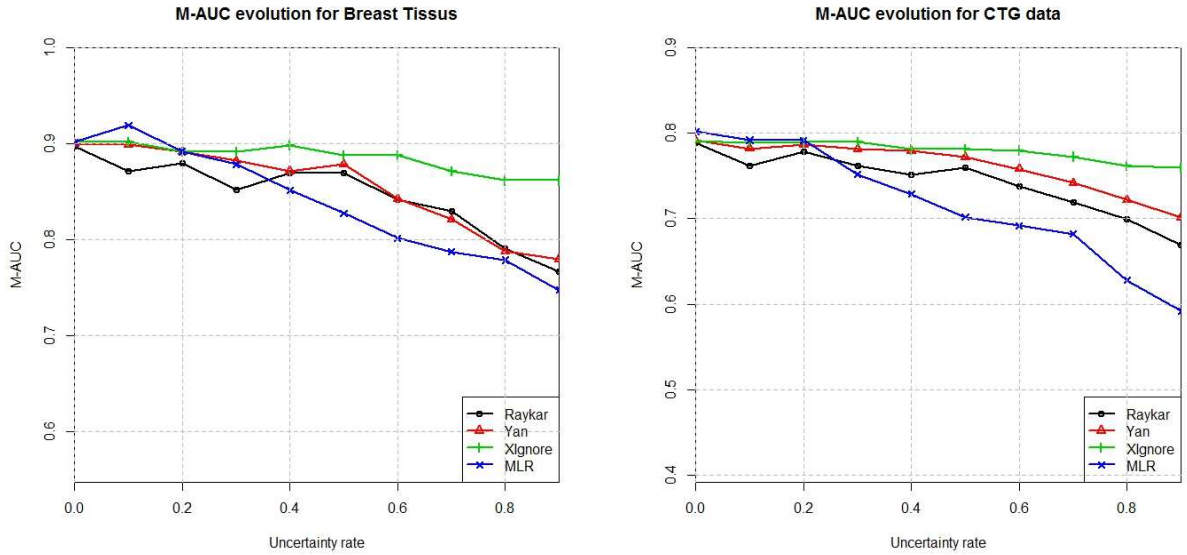


FIGURE 5.4 – Classification Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

TABLE 5.3 – Classification Multiclasses et Incertitude Totale : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	Error rate across different uncertainty levels			
Id	MLR	Raykar	Yan	XIgnore
1	0.500 ± 0.006	0.475 ± 0.002	0.429 ± 0.002	0.407 ± 0.001
2	0.625 ± 0.005	0.529 ± 0.001	0.518 ± 0.001	0.493 ± 10^{-4}
3	0.350 ± 0.008	0.305 ± 0.003	0.288 ± 0.002	0.350 ± 0.001
4	0.232 ± 0.001	0.242 ± 0.001	0.221 ± 0.001	0.203 ± 10^{-4}
5	0.146 ± 0.012	0.112 ± 0.009	0.097 ± 0.006	0.085 ± 0.001
6	0.155 ± 0.02	0.126 ± 0.005	0.106 ± 0.005	0.079 ± 0.003
7	0.595 ± 0.02	0.549 ± 0.008	0.548 ± 0.007	0.519 ± 0.004
8	0.241 ± 0.015	0.210 ± 0.006	0.196 ± 0.011	0.118 ± 0.001
9	0.550 ± 0.007	0.516 ± 0.005	0.487 ± 0.003	0.461 ± 10^{-4}
10	0.250 ± 0.012	0.169 ± 0.005	0.164 ± 0.007	0.107 ± 0.001
11	0.199 ± 0.002	0.164 ± 0.001	0.161 ± 0.001	0.140 ± 10^{-4}
12	0.331 ± 0.02	0.260 ± 0.007	0.268 ± 0.007	0.241 ± 0.003
Mean	$\mathbf{0.347 \pm 0.010}$	$\mathbf{0.304 \pm 0.004}$	$\mathbf{0.290 \pm 0}.004$	$\mathbf{0.266 \pm 0.001}$

X-Ignore Multiclasses et Incertitude Partielle

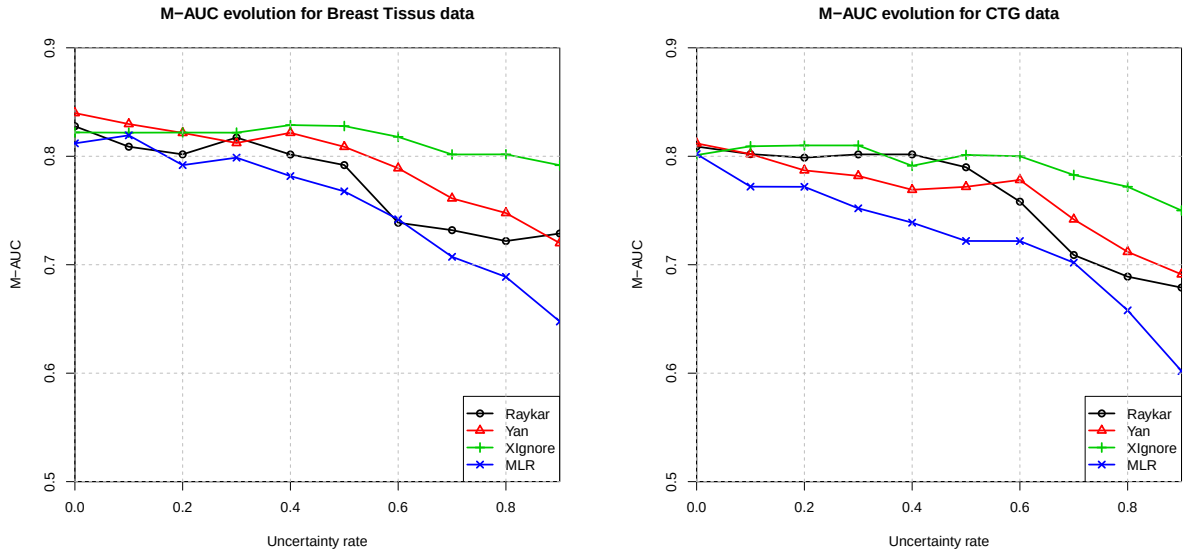


FIGURE 5.5 – Classification Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

TABLE 5.4 – Classification Multiclasses et Incertitude Partielle : Estimation en moyenne du taux d'erreur de classification et de la variance à travers tous les niveaux d'incertitude

	Error rate across different uncertainty levels			
Id	MLR	Raykar	Yan	XIgnore
1	0.450 ± 0.010	0.425 ± 0.011	0.400 ± 0.002	0.350 ± 0.002
2	0.550 ± 0.011	0.498 ± 0.009	0.496 ± 0.008	0.401 ± 10^{-4}
3	0.335 ± 0.009	0.321 ± 0.007	0.315 ± 0.008	0.290 ± 0.003
4	0.280 ± 0.002	0.248 ± 0.001	0.255 ± 0.003	0.195 ± 10^{-4}
5	0.250 ± 0.015	0.198 ± 0.010	0.220 ± 0.009	0.200 ± 0.001
6	0.230 ± 0.023	0.203 ± 0.019	0.198 ± 0.022	0.180 ± 0.002
7	0.459 ± 0.031	0.387 ± 0.020	0.401 ± 0.019	0.354 ± 0.003
8	0.230 ± 0.033	0.215 ± 0.025	0.210 ± 0.021	0.190 ± 0.010
9	0.220 ± 0.070	0.200 ± 0.050	0.230 ± 0.008	0.182 ± 10^{-4}
10	0.276 ± 0.021	0.156 ± 0.003	0.142 ± 0.004	0.102 ± 0.008
11	0.157 ± 0.004	0.110 ± 0.002	0.099 ± 0.003	0.099 ± 10^{-4}
12	0.329 ± 0.010	0.199 ± 0.005	0.200 ± 0.014	0.157 ± 0.005
Mean	$\mathbf{0.313 \pm 0.019}$	$\mathbf{0.263 \pm 0.013}$	$\mathbf{0.263 \pm 0.010}$	$\mathbf{0.225 \pm 0.002}$

5.4 Conclusion

Nous avons présenté dans ce chapitre la seconde contribution de cette thèse, à savoir le modèle X-Ignore, modèle générant un classifieur dans un contexte de classification supervisée en présence d'annotateurs multiples. Ce modèle vient en continuité du modèle Ignore développé au chapitre précédent, puisqu'il intègre aussi l'incertitude des juges lors de la génération du modèle. Cependant, contrairement à Ignore, l'originalité de X-Ignore vient du fait qu'il suppose que la fiabilité des annotateurs dépend non seulement de leur incertitude, mais aussi de la qualité des données à étiqueter. En effet, dans beaucoup de cas, la qualité des labels récoltée est considérablement liée à la qualité du jeu de données de départ. Ainsi, X-Ignore est une approche probabiliste bayésienne répondant au problème de classification supervisée en présence de multiples annotateurs, intégrant simultanément l'incertitude des juges et la qualité des données.
Les expérimentations effectuées sur de nombreux jeux de données de l'UCI ont montré l'efficacité et la stabilité de X-Ignore, comparée aux modèles baselines précédemment présentés.

Ignore et X-Ignore sont deux approches qui estiment la compétence des annotateurs lors de la génération du classifieur. Il peut cependant être intéressant de générer un classifieur effectuant une sélection des juges simultanément. En effet, avec l'utilisation des services d'annotations en ligne, les individus étiquetant les données peuvent avoir des niveaux de performance très hétérogène, et de nombreux annotateurs peuvent être incompétents face au problème posé. Or ces derniers augmentent le coût d'acquisition des labels et dégradent la qualité du classifieur. Il serait alors intéressant de générer un processus les éliminant afin d'augmenter la qualité du classifieur généré. C'est ce que nous allons étudier dans la 3ème contribution de cette thèse.

Chapitre 6

Sélection des ExpertS

Sommaire

Résumé : Le développement récent du crowdsourcing permet d'obtenir de manière simple et rapide des jeux de données étiquetés par de multiples annotateurs. Néanmoins l'utilisation de ces web-services a de grandes chances d'engendrer des annotations dominées par des spammers, i.e, des annotateurs de mauvaise qualité, dont le seul objectif est d'être rémunéré. Or les spammers augmentent significativement le coût d'acquisition des labels et dégradent la qualité du classifieur généré. Par conséquent, un mécanisme qui les détecte et les élimine est très clairement favorable lors de la génération du modèle.

Ce chapitre présente une nouvelle approche pour l'apprentissage à partir d'annotateurs multiples, dans le cas où les annotations sont dominées par des spammers. Notre algorithme ExpertS évalue les annotateurs, élimine les spammers, puis construit le modèle basé uniquement sur les annotateurs sélectionnés (experts). La particularité de ExpertS est la combinaison de deux métriques, l'entropie et le Spammer Score, pour la sélection des annotateurs.

On compare l'efficacité et la performance d'ExpertS par rapport à l'algorithme SpEM, algorithme récemment développé dans [Raykar and Yu, 2012] pour répondre au problème de classification et d'élimination des spammers en présence de multiples annotateurs. On montre que la combinaison des deux métriques dans ExpertS facilite et accélère considérablement l'étape d'élimination des spammers par rapport à SpEM. ExpertS a été publié dans [Wolley and Quafafou, 2013b].

6.1 Modèle Baseline de [Raykar et al., 2010]

Le système ExpertS que nous proposons ainsi que le système SpEM développé dans [Raykar and Yu, 2012] sont tous deux inspirés du modèle baseline de Raykar et Al. [Raykar et al., 2010]. Dans le but d'introduire les notations utilisées tout au long de ce chapitre, on rappelle ci-dessous certains points importants de leur travail.

Soit $X = \left[x_1^T; ...; x_N^T\right] \in R^{N \times D}$ la matrice composée des N instances x_i du jeu de données, $Y = \left[y_1^{(1)}, ..., y_1^{(T)}; ...; y_N^{(1)}, ..., y_N^{(T)}\right] \in R^{N \times T}$ la matrice des labels des annotateurs, et $Z = [z_1, ..., z_N]^T$ le vecteur des réels labels (inconnu). On se place dans un contexte de classification binaire. Soit $p = P[z_i = 1]$ la prévalence de la classe 1, en d'autres termes, la probabilité qu'une instance appartienne à cette classe. L'objectif est alors de maximiser la distribution conditionnelle jointe $P(Y, Z|\Theta)$, où Θ représente l'ensemble des paramètres à estimer du modèle. La maximisation suivant les paramètres Θ se fera à l'aide de l'algorithme EM, permettant finalement l'estimation de la probabilité $P(Z|Y, \Theta)$ à l'étape E.

En supposant que les instances (resp. les annotateurs) sont indépendantes (resp. indépendants) entre elles (resp. entre eux), et en décomposant suivant les valeurs possibles de z, les auteurs posent :

$$P[Y|\Theta] = \prod_{i=1}^{N} \prod_{t=1}^{T} P[y_i^t|z_i = 1, \Theta] \times P[z_i = 1|\Theta] + P[y_i^t|z_i = 0, \Theta] \times P[z_i = 0|\Theta]$$

Or dans le cas d'une classification binaire, $P[z_i = 0|\Theta] = 1 - P[z_i = 1|\Theta]$. De plus, les probabilités $P[y_i^t|z_i = 1, \Theta]$ et $P[y_i^t|z_i = 0, \Theta]$ peuvent être respectivement définies en terme de sensibilité α (taux de vrais positifs) et spécificité β (taux de vrais négatifs). On a alors :

$$P[Y|\Theta] = \prod_{i=1}^{N} \prod_{t=1}^{T} a_i p + b_i(1 - p)$$

où les auteurs définissent $a_i = \prod_{t=1}^{T} [\alpha^t]^{y_i^t} [1 - \alpha^t]^{1-y_i^t}$ et $b_i = \prod_{t=1}^{T} [\beta^t]^{1-y_i^t} [1 - \beta^t]^{y_i^t}$.

Les paramètres à estimer sont finalement les paramètres $\Theta = \left[\alpha^1, \beta^1, ..., \alpha^T, \beta^T, p\right]$. L'estimation de ces paramètres se fait à l'aide de l'algorithme EM, qui maximise le

logarithme de la vraisemblance :

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

A l'Etape E, la probabilité $\mu_i = P[z_i = 1|y_i^1, ..., y_i^T, \Theta]$ est estimée en calculant :

$$\begin{aligned} \mu_i &\propto P[y_i^1, ..., y_i^T|z_i = 1, \Theta]P[z_i = 1|\Theta] \\ &= \frac{a_i p}{a_i p + b_i(1 - p)} \end{aligned} \tag{6.1}$$

A l'Etape M, les paramètres Θ sont estimées en maximisant l'espérance conditionnelle :

$$E\left[ln\ P\left[Y, Z|\Theta\right]\right] = \sum_{i=1}^{N} \mu_i ln(pa_i) + (1 - \mu_i) ln(1 - p) b_i \tag{6.2}$$

La maximisation de cette expression suivant les différents paramètres donne les estimations suivantes :

$$\alpha^t = \frac{\sum_{i=1}^{N} \mu_i y_i^t}{\sum_{i=1}^{N} \mu_i} \tag{6.3}$$

$$\beta^t = \frac{\sum_{i=1}^{N} (1 - \mu_i)(1 - y_i^t)}{\sum_{i=1}^{N} (1 - \mu_i)} \tag{6.4}$$

$$p = \frac{\sum_{i=1}^{N} \mu_i}{N} \tag{6.5}$$

A partir de ce modèle, l'algorithme SpEM, publié dans [Raykar and Yu, 2012], propose d'intégrer une étape de sélection des annotateurs lors de la génération du classifieur. La prochaine section revient sur cette approche.

6.2 Méthode SpEM de [Raykar and Yu, 2012]

Jusqu'à aujourd'hui, très peu d'algorithmes combinent la classification en présence de multiples annotateurs et l'élimination des spammers. Une approche récente a tout de même était proposée par [Raykar and Yu, 2012], proposant l'algorithme SpEM qui fait appel à une méthode bayésienne pour la sélection des annotateurs.
Les auteurs introduisent le *spammer score*, correspondant à un score qui favorise les spammers, puis appliquent un a priori sur ces derniers pour les éliminer. La section qui suit revient sur la notion de *spammer score*.

6.2.1 Spammer Score

Suite aux études proposées dans [Raykar and Yu, 2011, Raykar and Yu, 2012], les auteurs modélisent la performance d'un annotateur pour chaque classe séparément. Si le réel label est la classe un, la sensibilité (taux de vrai positifs) pour l'annotateur t correspond à la probabilité que ce dernier annote un, i.e :

$$\alpha^t = P[y^t = 1|z = 1]$$

Réciproquement, si le réel label est zéro, la spécificité (vrais négatifs) est la probabilité que l'annotateur t annote bien la classe zéro, i.e :

$$\beta^t = P[y^t = 0|z = 0]$$

Un annotateur est alors défini comme étant un *expert* si sa sensibilité et sa spécificité sont proches de un. En d'autres termes, un annotateur est un expert si :

$$\alpha^t + \beta^t - 1 = 1 \tag{6.6}$$

A l'inverse, un *spammer* est défini comme étant un annotateur qui étiquette les instances aléatoirement, c'est-à-dire, indépendamment de l'instance x observée. On peut donner l'exemple d'un annotateur qui n'aurait pas compris les règles pour annoter le jeu de données, ou encore d'un annotateur ne regardant pas les instances en annotant (par manque de temps par exemple). Dans ce contexte, la probabilité qu'il annote une instance un ou zéro ne dépend pas du réel label de l'instance. En d'autres termes, un annotateur est un spammer si :

$$P[y^t = 1|z^t = 1] = P[y^t = 1|z^t = 0] \tag{6.7}$$

$$\alpha^t = 1 - \beta^t$$

$$\alpha^t + \beta^t - 1 = 0 \tag{6.8}$$

Or l'équation $\alpha^t = 1 - \beta^t$ correspond en réalité à la diagonale de la courbe ROC, puisque cette dernière admet en abscisse la valeur (1-spécificité) et en ordonnées la sensibilité.
Afin de mieux visualiser cette particularité, on simule 80 spammers et 20 experts pour le jeu de données binaire GLASS, disponible sur l'UCI Machine Learning Repository [Asuncion and Newman, 2007]. On estime la sensibilité et la spécificité de chaque annotateur à l'aide du modèle de [Raykar et al., 2010] décrit à la section 6.1. Les résultats sont représentés sur la Figure 6.1. On remarque que tous les spammers sont bien situés sur la diagonale du graphique.

Par conséquent, afin d'évaluer les annotateurs, [Raykar and Yu, 2012] définissent le Spammer Score S pour chaque annotateur t comme étant :

$$S^t = (\alpha^t + \beta^t - 1)^2 \tag{6.9}$$

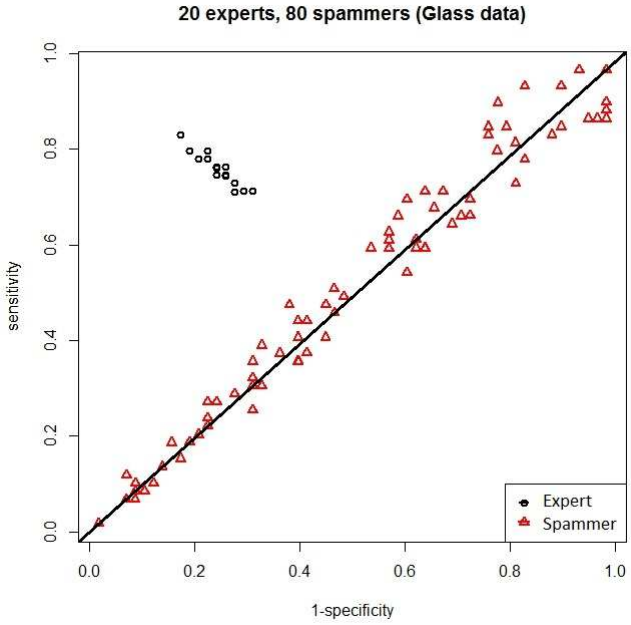


FIGURE 6.1 – Représentation des Annotateurs suivant leur Sensibilité et (1-spécificité) : 100 annotateurs simulés, 80 spammers (diagonale) et 20 experts.

Un juge est un spammer si S^t est proche de zéro. A l'inverse, il est perçu comme étant un expert si S^t est proche de un. En d'autres termes, on note $A = \{1, ..., T\}$ l'ensemble des annotateurs. Soit E l'ensemble des experts parmi les annotateurs. On a :

$$E = \left\{ t \in A | S^t > \phi \right\} \tag{6.10}$$

où $\phi \in [0, 1]$ est un seuil à fixer.

A présent que le *spammer score* est défini, on présente ci-dessous l'algorithme SpEM en détails.

6.2.2 Algorithme SpEM

[Raykar and Yu, 2012] utilise un a priori ASD (Automatic Detection of Spammer), basé sur le *spammer score*, pour favoriser les spammers et les éliminer par la suite. Plus précisement, l'a priori ASD de paramètre λ est fixé sur les paramètres de performances des annotateurs α et β, et s'écrit comme suit :

$$P[\alpha^t, \beta^t | \lambda^t] = \frac{1}{N(\lambda^t)} exp\left(-\frac{\lambda^t(\alpha^t + \beta^t - 1)^2}{2} \right)$$

Il propose ensuite de maximiser le logarithme a posteriori à l'aide de l'estimateur de maximum a posteriori (MAP) :

$$\hat{\Theta}_{max} = argmax \{ln\ P(Y, Z|\Theta) + ln\ P[\Theta]\}$$

avec $\Theta = \left[\alpha^1, \beta^1, ..., \alpha^T, \beta^T, p\right]$. L'a priori ASD engendre donc de nouveaux paramètres λ, qu'il est très important de bien fixer afin de ne pas pénaliser les bons annotateurs. Ce paramètre est estimé à l'aide d'une stratégie bayésienne (Maximum de Vraisemblance de Type II). L'algorithme SpEM est résumé dans l'algorithme. 8.

Algorithm 8 SpEM

Données de départ : Annotations $y_i^t, t = 1, ..., T, i = 1, ..., N$.
Initialiser $\lambda^t = 1/N$, pour t=1, ..., T.
Initialiser $\mathcal{A} = \{1, ..., T\}$ le groupe des bons annotateurs.
Initialiser $\mu_i = 1/T \sum_{t=1}^{T} y_i^t$ à l'aide du majority voting.
repeat
  repeat
    Recalculer les paramètres p, $\alpha^t, \beta^t,\ \forall t \in \mathcal{A}$, à l'aide de l'algorithme EM.
  until Convergence
  for all t $\in \mathcal{A}$ **do**
    Recalculer λ à l'aide du maximum de vraisemblance type II.
    if $\lambda^t > \delta_1$ (un seuil) **then**
      $\mathcal{A} \leftarrow \mathcal{A}| \{t\}$
    end if
  end for
until changement de l'a posteriori estimé $< \delta_2$
return α, β, p et les spammers détectés dans l'ensemble $\{1, ..., T\}|A$
Estimer le réel label de chaque instance du jeu de données à l'aide du Majority Voting sur l'ensemble des annotateurs sélectionnés.

Bien que l'algorithme SpEM soit efficace, il nous semble qu'estimer les paramètres pour l'ensemble des annotateurs peut devenir très vite contraignant et prendre énormément de temps. En effet, avec l'utilisation des web-services, on peut avoir un très grand nombre d'annotations, et le calcul des paramètres pour tous les annotateurs s'avère alors être très lourd et peut prendre énormément de temps. Ainsi, pour répondre à ce problème, on propose de combiner le spammer score avec la mesure de l'entropie, dans le but de réduire significativement le jeu de données à moindre coût.

6.3 Méthode ExpertS

Avant de procéder au développement de notre approche ExpertS, on introduit une définition de la mesure d'entropie.

6.3.1 Mesure Entropie

La notion d'entropie a été pour la première fois introduite par [Hartley, 1928], mais n'a été développée et utilisée qu'en 1949 par [Shannon and Weaver, 1949] dans le milieu industriel. Ces derniers ont proposé une mesure afin de quantifier l'information, correspondant aujourd'hui à la définition générale de l'entropie d'une distribution de probabilités. Le point clé de l'entropie, dans la théorie de l'information, est qu'elle mesure l'aléa d'une distribution ou d'un ensemble de variables observées. Il existe de multiples méthodes pour le calcul de l'entropie, celle de Shannon étant la plus connue et la plus utilisée d'entre elles [Shannon, 1948]. Elle est définie de la façon suivant :

Definition 8. *Soit X un ensemble de n variables aléatoires* $\{x_1, x_2, ..., x_n\}$. *L'entropie de Shannon, notée H(X), est définie comme suit :*

$$H(X) = -\sum_{i=1}^{n} p(x_i) ln(p(x_i)) \tag{6.11}$$

où $p(x_i)$ *est la probabilité d'occurence de la variable* x_i.

D'après l'entropie de Shannon, H(X) correspond à une mesure de l'aléa de la variable X. Si X suit une distribution uniforme, la valeur de son entropie H est maximale, ce qui indiquera un grand aléa sur la variable. Par conséquent, X détiendra très peu d'informations. A l'inverse, si l'entropie est faible (par exemple, pour une constante, on a H(X)=0), la variable détiendra beaucoup plus d'informations et sera donc considérée comme plus importante.

En apprentissage statistique, et plus précisément en classification supervisée, le calcul de l'entropie permet d'évaluer la quantité d'information qu'apporte une variable par rapport au problème considéré. Ainsi, cette notion peut être utilisée dans le but de réduire la dimension d'un jeu de données.
Dans le cas d'une classification en présence d'annotateurs multiples, le calcul de l'entropie peut alors s'avérer être une solution afin de sélectionner les experts et d'éliminer les spammers. Néanmoins, la mesure de l'entropie telle qu'elle est définie par Shannon n'est pas toujours valide dans ce contexte. En effet, prenons le cas d'un jeu de données binaire où les classes sont équilibrées, c'est-à-dire en présence d'un nombre équivalent d'instances pour chaque classe. Dans ce contexte, un expert annotera environ autant de fois le label 0 que le label 1, puisqu'il attribue la plupart du temps le bon label à chaque instance. La distribution de ses labels sera donc uniforme, avec environ autant de labels 1 que de labels 0. Cela conduit finalement à une valeur élevée de son entropie. A l'inverse, un spammer ayant par exemple toujours annoté 1 ou 0 aura une faible valeur d'entropie.
A présent, prenons le cas d'un jeu de données avec des classes déséquilibrées, c'est-à-dire où une classe est majoritaire par rapport à l'autre. Dans ce contexte, l'entropie d'un annotateur expert serait faible, puisqu'il aurait donné un label beaucoup plus souvent qu'un autre. A l'inverse, un spammer qui aurait annoté les instances aléatoirement aurait

une entropie beaucoup plus élevée.
Le calcul de l'entropie suivant la définition de Shannon ne nous permet pas d'identifier les spammers et les experts, puisque cette dernière suppose que la distribution uniforme est toujours la distribution la plus imprécise et celle possédant le moins d'information. Or nous venons de voir que cela n'est pas toujours le cas en apprentissage.

Une nouvelle entropie possédant des propriétés plus appropriées a alors été introduite par [Zighed et al., 2010]. Soit H_w la nouvelle entropie. H_w est définie comme suit :

$$H_w(N, f_1, f_2, ..., f_n) = \sum_{i=1}^{n} \frac{\lambda_i(1-\lambda_i)}{(-2w_i+1)\lambda_i + w_i^2} \tag{6.12}$$

avec :

$$\lambda_i = \frac{Nf_i + 1}{N + n} \tag{6.13}$$

f_i correspond à la fréquence de chaque classe i, et w_i a une distribution dite de "référence", qui est perçue comme étant la distribution à entropie maximale. Cette nouvelle distribution ne considère plus la distribution uniforme comme la distribution la plus imprécise, mais considère à la place une distribution de "référence" correspondant à la probabilité d'occurence de chaque classe dans le jeu de départ.
Cette entropie est donc plus appropriée dans le cadre de classification, puisqu'elle dépend de la distribution des véritables labels dans le jeu de données initial.
La question que l'on se pose alors est de savoir comment déterminer cette distribution de référence. Dans le cas où la probabilité d'occurence de chaque classe est connue au départ, il est naturel de se servir de ces probabilités a priori dans le but de déterminer la distribution w_i. Dans le cas contraire, elle peut être estimée en calculant la moyenne des fréquences pour chaque classe, dans le jeu d'apprentissage.
De cette manière, lorsque de multiple annotateurs sont présents, les experts, dont la distribution des labels sera a peu près équivalente à la distribution réelle des classes dans le jeu de données, auront l'entropie la plus élevée. Le graphique 6.2 montre les résultats du calcul d'entropie H_w pour chaque annotateur simulé dans la section 6.2.1.

On remarque que contrairement aux experts, la valeur de l'entropie des spammers varie énormément. Ce résultat est logique, puisque par définition, un spammer est un juge annotant le jeu de données aléatoirement. De ce fait, la distribution des labels résultante est aussi aléatoire et peut être équivalente à la distribution de référence.

On note H_w^t la distribution de l'annotateur t. On a $H_w = \left\{H_w^1, ..., H_w^T\right\}$. On note EC le groupe des Experts Candidats représentant les annotateurs les plus probables d'être des experts. On a alors :

$$EC = \left\{t \in A | H_w^t > \lambda\right\} \tag{6.14}$$

où $\lambda \in [0, +\infty[$. En d'autres termes, si l'entropie d'un annotateur est supérieur à ce seuil λ, ce dernier est probablement un expert. Dans le cas contraire, l'annotateur aura de

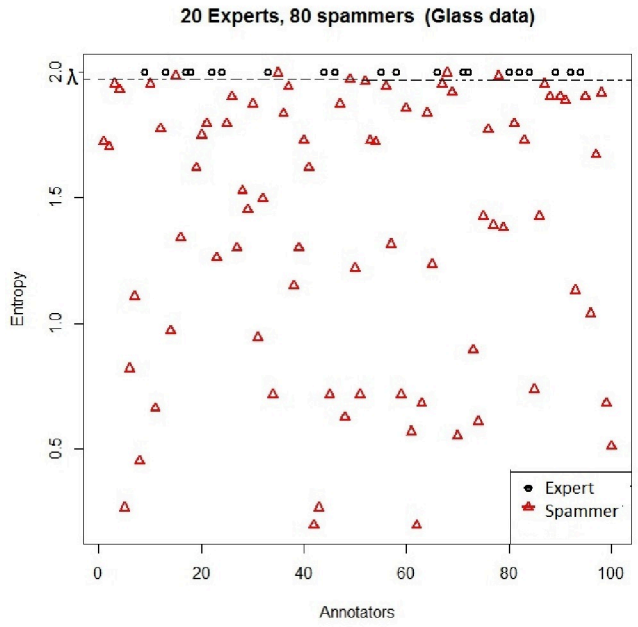


FIGURE 6.2 – Représentation de la mesure de l'entropie pour chaque annotateur simulé.

fortes chances d'être un spammer. Le choix de la valeur du paramètre λ sera étudié dans la prochaine section.

L'utilisation de la mesure d'entropie est une très bonne méthode pour éliminer un large groupe de spammers et réduire ainsi les annotations obtenues par les web-services. Combinée avec le spammer score défini à la section 6.2.1, on obtient la méthode ExpertS, un algorithme efficace et peu coûteux en terme de calculs, qui estime la performance des annotateurs, élimine les spammers, et génère le classifieur en se basant uniquement sur les annotations des experts.
La prochaine section décrit plus précisément les étapes de l'algorithme ExpertS.

6.3.2 Algorithme ExpertS

Comme nous l'avons décrit dans la section précédente, l'étape de sélection des experts est établie à l'aide d'une combinaison de deux scores : la mesure d'entropie et le spammer score. Pour commencer, on calcule l'entropie pour chaque annotateur dans le but d'identifier le groupe EC, puis l'algorithme EM est utilisé sur le groupe sélectionné pour estimer la performance de chaque annotateur et éliminer les spammers dissimulés dans ce groupe. Ainsi, dans ce travail, on définit un expert comme étant un annotateur satisfaisant les deux équations 6.10 et 6.14. Soit E_f l'ensemble des experts sélectionnés

à la fin. On a :

$$E_f = \{t \in (EC \cap E)\} \tag{6.15}$$

$$= \Big\{t \in A | (H_w^t > \lambda) \cap (S^t > \phi)\Big\} \tag{6.16}$$

où $\lambda \in [0, +\infty[$ et $\phi \in [0, 1]$.

Suite à de multiples expérimentations, dans ce travail on suppose que les bons annotateurs ont une sensibilité α et une spécificité β supérieure à 0.6. On appelle alors spammers les annotateurs t qui satisfont $S^t \leq 0.04$ (d'après l'équation (6.9)), et on les filtre durant la procédure d'élimination. Par conséquent, dans nos expérimentations, on pose $\phi = 0.04$.
Il reste maintenant à fixer la valeur du second seuil λ. Or la seule donnée que nous possédons concernant ce seuil est que sa valeur est dans l'intervalle $[0, +\infty[$. Comment peut-on alors identifier le groupe EC ? On propose ici d'appliquer une méthode à trois étapes :

1. Etape de Classement : On commence par calculer l'entropie H_w^t pour chaque annotateur, et on définit $(A, \leq)$ la structure qui les ordonne selon la valeur de leur entropie dans un ordre croissant. On fixe le groupe EC comme étant le groupe avec les K annotateurs possédant les entropies les plus élevées.
2. Etape d'Evaluation : On utilise l'algorithme EM sur le groupe EC dans le but d'estimer la sensibilité et la spécificité de chaque annotateur.
3. Etape de filtrage : On calcule le *spammer score* S^t pour chaque annotateur, et on défini le groupe final des experts comme suit :

$$E_f = EC - \Big\{t \in EC | S^t \leq \phi\Big\}$$

On répète les étapes 1, 2, et 3 en ajoutant, à chaque itération, les prochains Top-K annotateurs au groupe final des experts sélectionnés, jusqu'à ce qu'il n'y ait plus d'experts trouvés. On appelle Top-K annotateurs les K annotateurs dont les valeurs d'entropie sont les plus élevées.

Pour finir, on utilise le majority voting pour prédire le véritable label pour chaque instance. L'algorithme ExpertS est résumé dans l'Algorithme 9.

6.4 Résultats Expérimentaux

On montre l'efficacité de la méthode proposée sur des données de UCI Machine Learning Repository [Asuncion and Newman, 2007]. Les experts ont été simulés avec une sensibilité et spécificité entre 0.75 et 0.95, alors que tous les experts sont situés autour de la diagonal du graphique ROC. On compare ExpertS avec trois méthodes : le classique majority voting, la méthode baseline de Raykar et Al.[Raykar et al., 2010], et la méthode SpEM développée dans [Raykar and Yu, 2012]. Parmi toutes ces méthodes,

Algorithm 9 ExpertS

Données de départ : X, Y.
Initialiser $\alpha = 0, \beta = 0$, seuils ϕ et le nombre K des annotateurs à ajouter à chaque itération.
Initialiser $A = \{1, ..., T\}$.
Calculer $H_w(A) = \left\{H_w^1, ..., H_w^T\right\}$.
Etape de classement : EC $\leftarrow TopK(A, \leq)$.
Estimer le véritable label pour chaque instance : $\mu_i = 1/|EC| \sum_{t \in EC} y_i^t$
Etape d'évaluation : $\forall t \in EC$, réinitialiser $\{\alpha^t, \beta^t\}$ et la prévalence p à l 'aide de l'algorithme EM et des équations 6.3, 6.4 et 6.5.
Etape de filtrage : Calculer $S^t, \forall t \in EC$. Sélectionner le groupe d'experts final :

$$E_f^{old} \leftarrow EC \setminus \left\{t \in EC | S^t \leq \phi\right\}$$

.
repeat
  A $\leftarrow$ A$\setminus TopK(A, \leq)$.
  EC $\leftarrow TopK(A, \leq) \cup E_f^{old}$.
  Recalculer $\mu_i = 1/|EC| \sum_{t \in EC} y_i^t$
  Algorithme EM : Recalculer $\{\alpha^t, \beta^t, p\}$ à l'aide des équations 6.3, 6.4 et 6.5.
  Calculer $S^t,\ \forall t \in EC$. Sélectionner le groupe d'experts final :

$$E_f^{new} \leftarrow EC \setminus \left\{t \in EC | S^t \leq \phi\right\}$$

  .
  Booléen $\leftarrow (E_{old}^f = E_{new}^f)$.
  if non Booléen **then**
    $E_f^{old} \leftarrow E_f^{new}$
  end if
until Booléen
return Paramètres α, β, p et l'ensemble des experts E_f^{new}
Prédire le véritable label pour chaque instance à l'aide du majority voting sur tous les experts sélectionnés.

seul l'algorithme SpEM a un mécanisme qui détecte explicitement les spammers et les élimine. On simule chaque modèle cent fois à l'aide de la méthode bootstrap, et on évalue notre algorithme en utilisant les deux critères suivant : l'AUC (Area Under Curve) pour estimer la précision de chaque modèle, et le temps d'exécution (en secondes) dans le but d'estimer leur rapidité.

6.4.1 Performance de l'Algorithme ExpertS

On s'intéresse dans un premier temps à valider la performance du classifieur généré par le modèle ExpertS. Ce critère est vérifié sur 8 jeux de données de l'UCI qui sont : Ionosphere (351,34), Cleveland Heart (297,13), Musk(version 1) (476,167), Glass (214,10), Bupa (345,7), Vertebral (310,6), Spect Heart (267,22) and Haberman (306,3) (avec (nombre d'instance, nombre de variables)). Le lecteur peut se réferer à la Table. 4.1 pour plus de précisions sur ces données. On simule 100 annotateurs, pour lesquels 15% sont des experts et 85% des spammers, et on fixe le seuil K à 0.30 fois le nombre d'annotateurs. On calcule pour chaque jeu de données et pour les 4 modèles générés l'AUC (cf.Table 6.1) ainsi que le temps d'exécution (cf.Table 6.2). En complément, on estime la sensibilité de la détection des spammers pour les deux algorithmes SpEM et ExpertS, correspondant à la fraction de spammers correctement détectée (cf.Table 6.3).

TABLE 6.1 – Comparaison de l'AUC pour ExpertS, Majority Voting (M.V), Baseline, et SpEM.

Dataset	M.V	Baseline	SpEM	ExpertS
Cleveland	0.922	0.934	0.991	0.998
Ionosphere	0.901	0.951	0.995	0.994
Musk	0.930	0.952	1.000	0.998
Glass	0.860	0.907	0.995	0.997
Bupa	0.822	0.876	0.989	0.990
Vertebral	0.900	0.930	0.998	0.996
Spect Heart	0.865	0.904	0.996	0.992
Haberman	0.721	0.875	0.997	1.000
Mean	0.865	0.916	0.995	0.996

A partir des résultats observés, on peut faire les différentes remarques ci-dessous :

1. Concernant la qualité du classifieur généré, on confirme la supériorité de SpEM et ExpertS comparée à la méthode baseline et au majority voting. L'AUC des deux algorithmes d'élimination des spammers est en effet plus élevé pour tous les jeux de données (se référer à la Table.6.1). Ce résultat montre l'efficacité de la sélection des bons annotateurs lors de la génération du classifieur. En complément, les résultats de la Table.6.3 confirment que ExpertS et SpEM sont aussi bons l'un que l'autre puisqu'ils se retrouvent tous les deux avec une sensibilité de 99% pour la détection des spammers.

TABLE 6.2 – Comparaison du Temps d'Exécution (en secondes) pour ExpertS, Majority Voting (M.V), Baseline, et SpEM.

Dataset	M.V	Baseline	SpEM	ExpertS
Cleveland	0.86	41.35	152.25	1.66
Ionosphere	1.02	97.77	137.83	3.31
Musk	1.26	157.87	203.63	5.07
Glass	0.59	20.56	62.70	1.56
Bupa	0.86	41.35	152.25	1.63
Vertebral	0.91	107.96	201.41	1.92
Spect Heart	0.91	67.30	134.69	1.72
Haberman	0.89	42.18	258.92	1.81
Mean	0.91	72.04	162.96	2.34

TABLE 6.3 – Comparaison du Taux de Spammers Correctement Détecté pour SpEM et ExpertS.

Dataset	SpEM	ExpertS
Cleveland	0.998	0.997
Ionosphere	0.996	0.995
Musk	0.995	0.996
Glass	0.999	0.989
Bupa	0.987	0.987
Vertebral	0.991	0.997
Spect Heart	0.987	0.999
Haberman	0.989	0.988
Mean	0.993	0.994

2. L'avantage de l'algorithme ExpertS par rapport à SpEM est observé à la Table.6.2, où on peut noter que l'algorithme proposé est beaucoup plus performant en terme de temps d'exécution, ce dernier étant significativement moins élevé (autour de 163sec pour SpEM comparé à 2.34sec pour ExpertS).

Pour résumé, l'algorithme ExpertS proposé dans ce chapitre est plus rapide et plus performant (ou aussi performant) que de précédentes approches, sans dégrader la qualité du modèle généré.

6.4.2 Effet de l'Augmentation du Nombre d'Annotateurs

Avec l'utilisation des web-services, il est possible de se retrouver avec des jeux de données de très grande dimension, incluant des centaines d'annotateurs. Ainsi, dans cette section, on teste la robustesse de notre méthode quand elle est confrontée à un grand nombre d'annotateurs. On simule pour commencer 200 annotateurs, puis on ajoute à

chaque itération 200 annotateurs, jusqu'à atteindre 10000 annotateurs. Pour chaque simulation, 15% sont des experts et 85% des spammers. On calcule l'AUC et le temps d'exécution (en secondes) pour chaque modèle. Les résultats obtenus pour le jeu de données Glass peuvent être vus sur les Figures 6.3 : pour tous les modèles testés, la qualité des labels estimée ne dépend pas du nombre d'annotateurs simulé : l'algorithme ExpertS que l'on propose ainsi que SpEM sont dans tous les cas meilleurs que les modèles baseline et majority voting. Néanmoins, contrairement à ExpertS, SpEM a clairement l'inconvénient d'avoir un temps d'exécution qui accroit linéairement en fonction du nombre d'annotateurs (le temps d'exécution de ExpertS varie de 1 seconde à 8 secondes, alors que SpEM varie de 1 seconde à 5000 secondes). Par conséquent, l'algorithme que l'on propose est plus pratique lorsque l'on est confronté à un nombre d'annotateurs élevé.

6.4.3 Effet de l'Augmentation du Nombre de Spammers

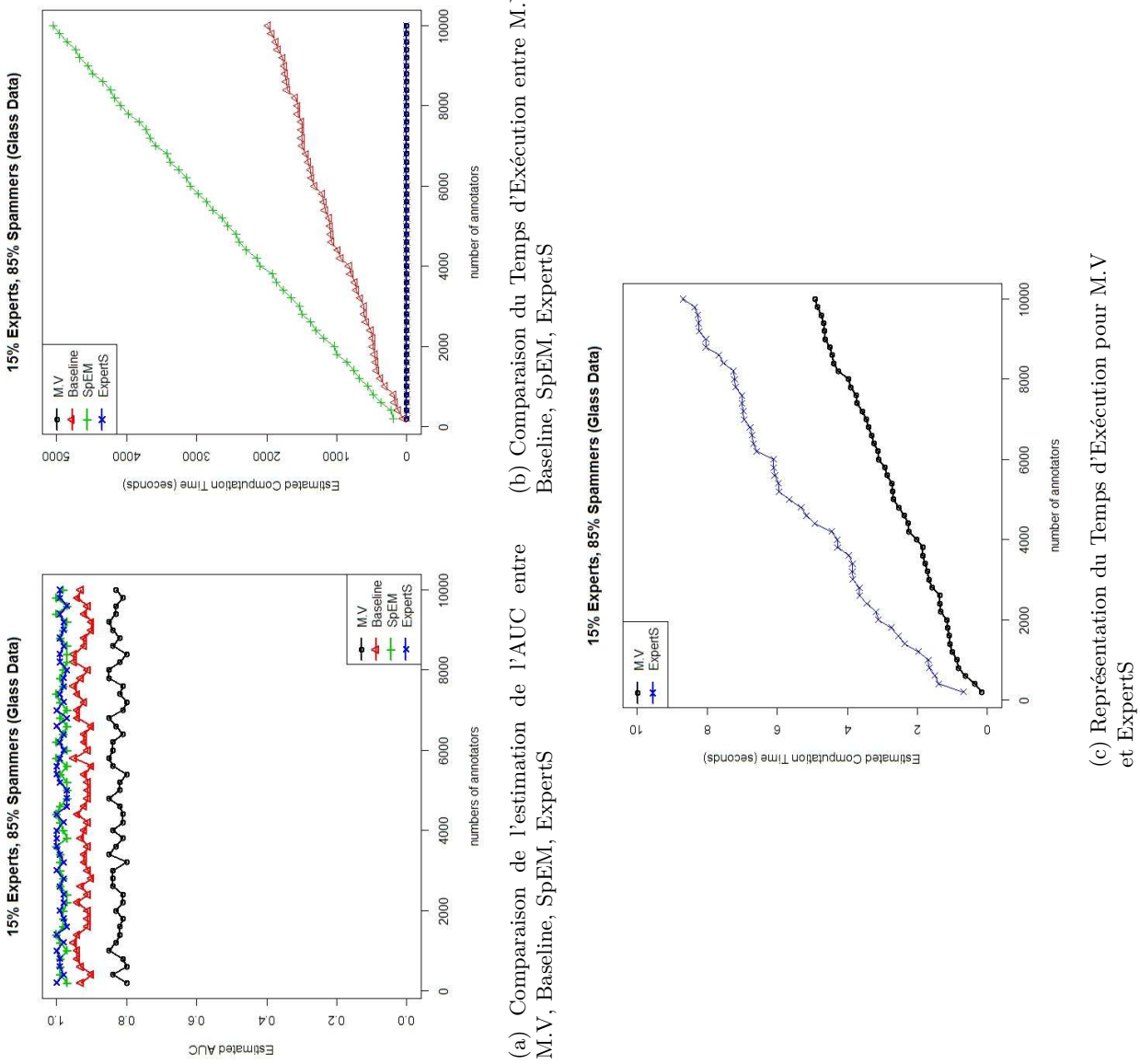
On étudie ici l'effet de l'augmentation du nombre de spammers parmi les annotateurs. On simule 5 experts pour le jeu de données Glass et on ajoute à chaque itération 100 spammers. On compare l'évolution de l'AUC obtenu, et on représente la sensibilité de la détection des spammers. En supplément, on représente le nombre d'annotateurs utilisé durant toutes les itérations, pour SpEM et ExpertS. Les résultats sont reportés sur les Figures 6.4. Dans un premier temps, on remarque que SpEM et ExpertS achèvent une meilleure performance et sont beaucoup plus robustes que le majority voting et que la méthode Baseline, lorsque le nombre de spammers est élevé. Ce résultat est très facilement expliqué par le fait que les deux algorithmes SpEM et ExpertS éliminent les spammers lors de la génération du classifieur. De plus, on remarque que le nombre d'annotateurs utilisé à travers toutes les itérations est significativement moins élevé dans notre algorithme que dans l'algorithme SpEM. Ce résultat explique la réduction du temps d'exécution de notre algorithme par rapport à SpEM, et valide notre approche.

6.4.4 Effet du seuil K

Comme décrit à la section 6.3.2, un paramètre K représentant le nombre d'annotateurs initial et à ajouter à chaque itération doit être fixé. Pour les précédentes expérimentations, ce seuil a été fixé à 0.30 fois le nombre d'annotateurs. Néanmoins, on peut utiliser ce paramètre afin de contrôler le nombre d'annotateurs à prendre en compte et à ajouter à chaque itération. La Figure.6.5 montre la performance de ExpertS pour différentes valeurs de K. On peut constater que ExpertS est optimisé en terme de précision et de temps d'exécution pour un seuil compris entre 0.30 et 0.60.

6.5 Conclusion

Dans ce chapitre, nous avons proposé la méthode ExpertS, correspondant à une approche probabiliste pour la classification supervisée en présence de multiples annotateurs. A la différence des deux modèles Ignore et X-Ignore précédemment développés, ExpertS a l'originalité de générer un classifieur tout en filtrant simultanément les annotateurs

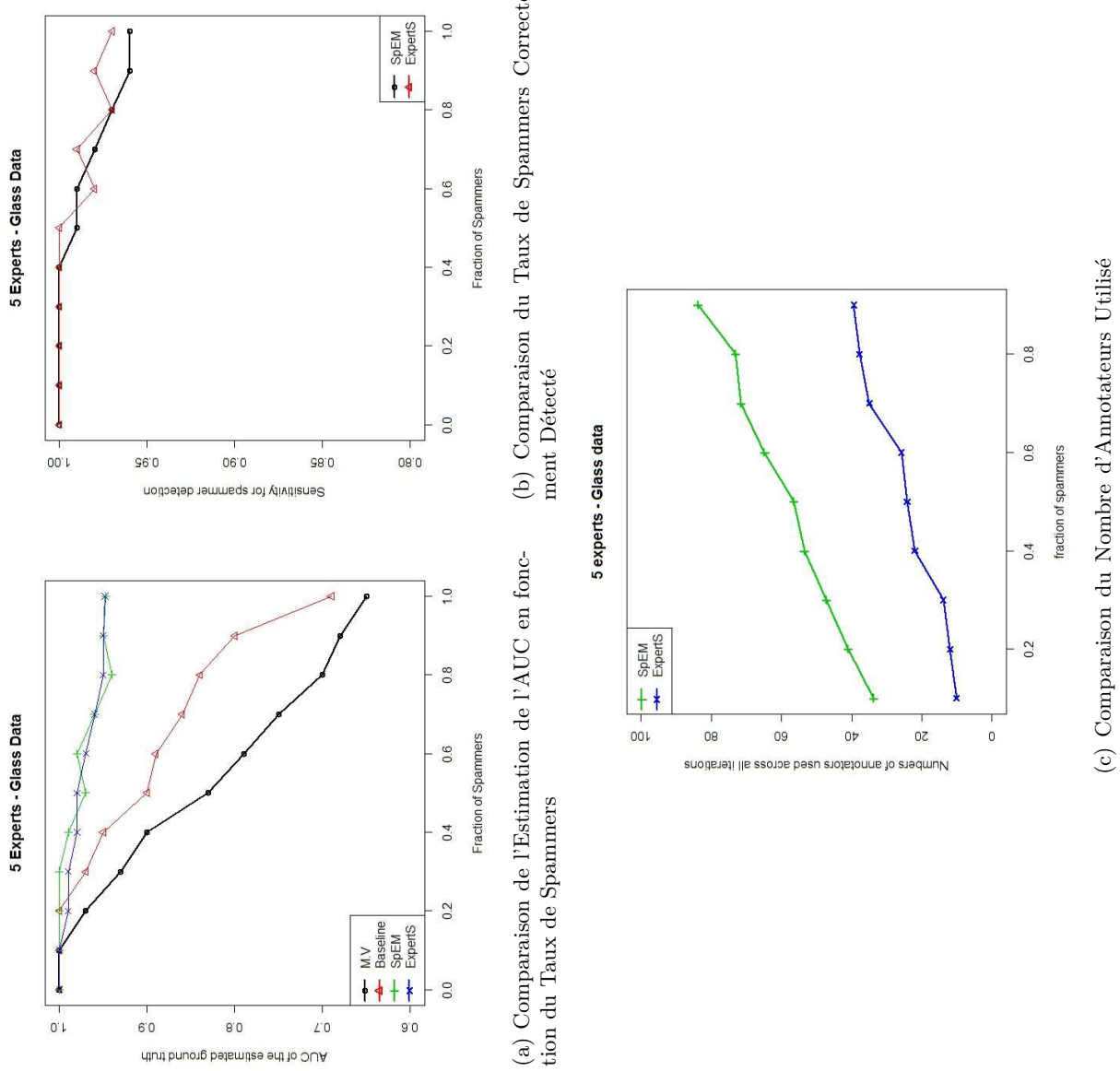


(a) Comparaison de l'estimation de l'AUC entre M.V, Baseline, SpEM, ExpertS

(b) Comparaison du Temps d'Exécution entre M.V, Baseline, SpEM, ExpertS

(c) Représentation du Temps d'Exécution pour M.V et ExpertS

FIGURE 6.3 – Effet de l'Accroissement du Nombre d'Annotateurs.

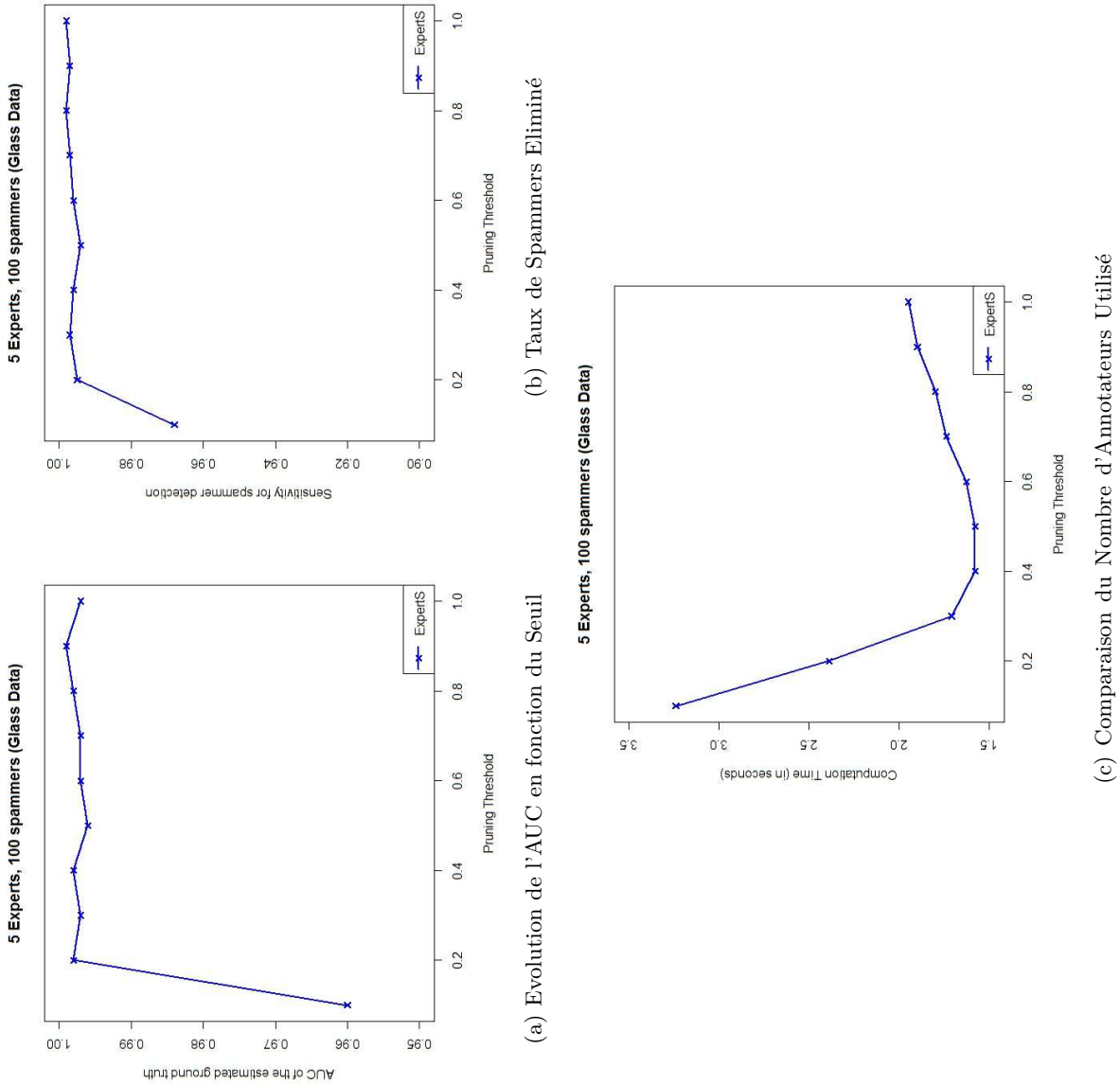


(a) Comparaison de l'Estimation de l'AUC en fonction du Taux de Spammers

(b) Comparaison du Taux de Spammers Correctement Détecté

(c) Comparaison du Nombre d'Annotateurs Utilisé durant toutes les Itérations

FIGURE 6.4 – Effet de l'Accroissement du Nombre de Spammers.



(a) Evolution de l'AUC en fonction du Seuil

(b) Taux de Spammers Eliminé

(c) Comparaison du Nombre d'Annotateurs Utilisé durant toutes les Itérations

FIGURE 6.5 – Temps d'Exécution (en secondes) en fonction de la Valeur du Seuil.

spammers et en ne sélectionnant que les experts. Ainsi, l'algorithme ExpertS élimine les juges non compétents pour ne générer le classifieur qu'en présence de labels de haute qualité.
La deuxième originalité de ExpertS est la combinaison de deux métriques, le spammer score et l'entropie, afin d'effectuer l'étape de sélection des annotateurs. L'utilisation de ces deux métriques aboutit à une rapidité du temps d'exécution de l'algorithme, même en présence de centaines d'annotateurs.
Les résultats des différentes expérimentations ont montré l'efficacité et la rapidité de ExpertS en comparaison avec d'autres algorithmes Baseline de référence, dont l'algorithme SpEM, algorithme éliminant également les spammers lors de la génération du classifieur.

Nous nous sommes jusqu'à présent restreints à tester la performance des trois approches proposées sur des données synthétiques. Or il serait à présent intéressant d'appliquer ces méthodes sur des données réelles. Ce sera l'objectif du chapitre qui suit.

Chapitre 7

Une Application Médicale Réelle : Mélanome

Sommaire

Résumé : L'objectif de ce chapitre est d'appliquer les 3 contributions proposées, à savoir les modèles Ignore, X-Ignore et ExpertS à un domaine réel. En effet, jusqu'à présent ces trois modèles ont été évalués sur des données synthétiques de l'UCI ; ce chapitre complète alors nos expérimentations en appliquant chacune des méthodes sur des données médicales réelles, et étudie les différents résultats obtenus.

7.1 Présentation du Cadre Applicatif

Le cancer de la peau représente le plus important cancer chez l'être humain, et son nombre ne cesse d'augmenter avec le temps. Parmi les nombreux types de cancers de la peau, le mélanome reste aujourd'hui l'un des plus graves, puisqu'il entraine la mort dans 75% des cas. Ainsi, de nombreux chercheurs ont étudié ce problème afin d'arriver à une solution pour détecter ce cancer le plus tôt possible chez les patients

[Gareau et al., 2012, Pereyra et al., 2012, Ballerini et al., 2012].

Nous allons alors évaluer les trois modèles Ignore, X-Ignore et ExpertS développés dans cette thèse sur une application médicale réelle, *Mélanome*, dont le but est de prédire, à partir de plusieurs caractéristiques décrivant des images de lésions cutanées, si l'on est en présence d'un nævus malin ou bénin. Le nævus est plus connu sous le nom de grain de beauté, et prend le nom de mélanome dans le cas où il est malin.
Dans un contexte d'apprentissage, *Mélanome* nous permet de générer un classifieur dans le cadre d'une classification supervisée binaire, où le réel label pour chaque image est malin ou bénin. Par ailleurs, 10 dermatologistes séniors ont été sollicités afin de donner leur avis sur l'état de chacun des patients suivant l'image de sa lésion cutanée correspondante. Nous présentons ci-dessous plus en détails le jeu de données collecté.

7.1.1 Description des Données

Le jeu de données *Mélanome* a été collecté via une application web entre janvier 2006 et septembre 2011. Il est initialement composé de 7000 images de lésions cutanées obtenues par un ensemble de 16 dermatologistes équipés d'une caméra SONY W120, combiné à un dermoscope. Un dermoscope est un appareil composé d'une ou de plusieurs lentilles grossissantes et d'un système d'éclairage permettant d'obtenir des observations plus précises de lésions de la peau. Un exemple de dermoscope peut être vu à la Figure 7.1.
Les motifs binaires locaux (Local Binary Pattern en anglais, LBP) [Harwood et al., 1995], représentent des caractéristiques très souvent utilisées en vision par ordinateur pour reconnaître des textures dans des images numériques. Le nombre total de différents LBP étant relativement restreint, (entre 256 et 512), les dermatologues décident d'étudier à la place les HLBP, correspondant au nombre d'occurrences de chaque LBP dans une image [Paris et al., 2012]. HLBP a l'avantage de pouvoir extraire relativement facilement des structures générales des régions d'intérêts des images, tout en étant moins sensible aux caractéristiques locaux. Cette propriété est importante lorsque l'on souhaite arriver à des conclusions les plus générales possibles. Ainsi, chaque image a été décomposée en de multiples zones décrites par les HLBP, menant finalement à ce que chacune d'elles soit décrite par un vecteur de dimension 30720. Récemment, 1097 images ont été utilisées dans la thèse de [Wazaefi, 2013] , et nous avons repris ces mêmes images dans notre étude. Ainsi, notre jeu de données est de taille 1097×30720. Parmi les 1097 images (donc 1097 patients), seules 88 ont été répertoriées ayant un nævus malin (mélanome), les autres patients ayant un nævus bénin. Ces deux classes correspondent dans notre étude aux réels labels des patients, représentés par un vecteur de dimension 1097. Notre étude se place ainsi dans un contexte de classification binaire, et utilise donc les versions binaires de nos modèles Ignore et X-Ignore.

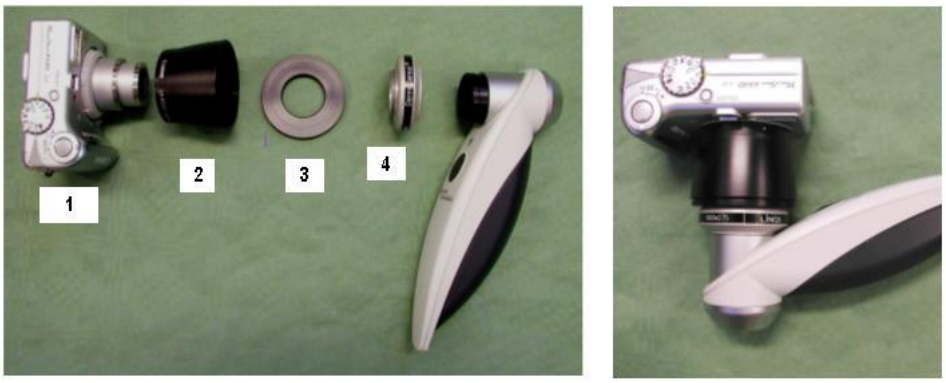


FIGURE 7.1 – Exemple de Dermascope Combiné à une Caméra.

7.1.2 Les Labels des Annotateurs

Via l'application web conçue pour collecter les images, 10 dermatologues séniors ont été sollicités afin d'annoter les 1097 images. Un exemple d'image de nævus bénin et malin peut être vu à la Figure 7.2. Les dermatologues n'avaient pas de limitation de temps pour annoter les images, et chaque image a été mise en ligne de manière à ce qu'elle soit de très bonne qualité, et à ce qu'ils puissent zoomer pour l'agrandir. Pour chaque image, les dermatologues ont la possibilité d'exprimer leur incertitude quant au label qu'ils donnent. Ainsi, ils peuvent répondre par "lésion certainement bénigne (resp. maligne)" dans le cas de certitude, ou "lésion probablement bénigne (resp. maligne)" dans le cas de doute. Finalement, les annotations des juges se présentent sous la forme d'une matrice de taille 10×1097. Notre étude se place ainsi dans un contexte de classification binaire, où les annotateurs expriment leur certitude ou leur doute, sans associer de taux d'incertitude à chacun de ces deux cas. On utilise donc les versions binaires avec incertitude totale de nos modèles Ignore et X-Ignore.

Dans le but de générer un classifieur pour prédire le label d'une nouvelle image, deux méthodes sont possibles. La première consiste à ne prendre en compte que la matrice des Images (1097×30720) ainsi que les réels labels (vecteur de taille 1097). On se retrouve alors dans un contexte de classification binaire classique, où le réel label est connu pour chaque instance dans le jeu de données initial. De très nombreuses méthodes existent pour répondre à ce problème (Régression Logistique, SVM) et c'est dans ce contexte que [Wazaefi, 2013] a mené son étude. La deuxième méthode consiste à supposer que les réels labels sont très difficiles à obtenir, et donc à les remplacer par la matrice d'annotations des juges. On se place alors dans un contexte de classification supervisée en présence de multiples annotateurs, où leur incertitude est prise en compte. Ce deuxième cadre d'étude correspond en réalité au contexte principal de cette thèse, où les trois modèles Ignore, X-Ignore et ExpertS ont été développées pour traiter ce problème. C'est dans ce contexte que nous pouvons alors (1) inférer un label (bénin ou malin) pour une nouvelle

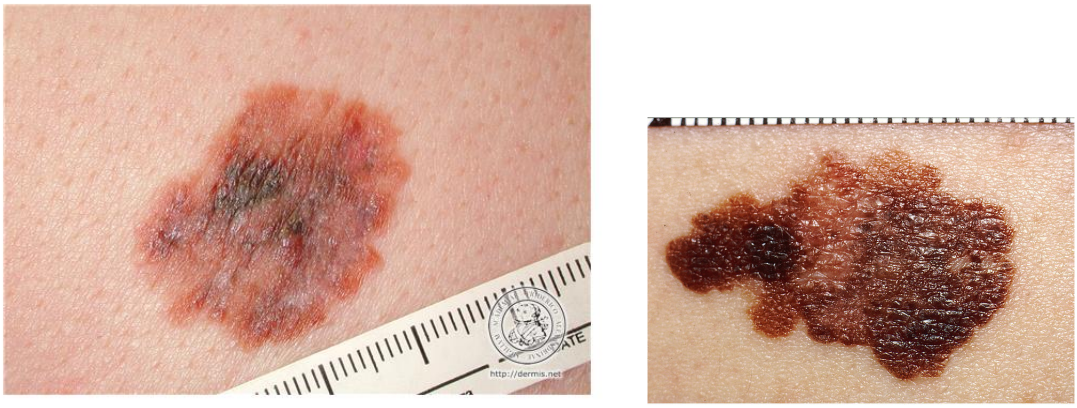


FIGURE 7.2 – Un Exemple d'Image de Nævus Bénin (à gauche) et Malin (Mélanome, à droite).

image, (2) évaluer les performances des annotateurs et (3) évaluer la difficulté d'une image à être annotée, et inférer un degré de difficulté pour une nouvelle image.

Comme nous l'avons souligné précédemment, le cadre de la classification binaire classique du jeu de données *Mélanome* a déjà fait l'objet d'une étude apparue dans la thèse de [Wazaefi, 2013]. La prochaine section revient sur les résultats obtenus à travers cette thèse, afin de les comparer plus tard dans le chapitre aux résultats dans le nouveau contexte de la génération du classifieur uniquement en se basant sur les annotations des juges.

7.2 Retour sur l'Etude de [Wazaefi, 2013]

Dans un contexte de classification supervisé binaire classique où le réel label est connu, [Wazaefi, 2013] a généré un classifieur pour *Mélanome*, dans le but de prédire les labels réels de nouvelles images de lésions cutanées. Or *Mélanome* étant un jeu de données de très grande dimension, il s'avère être important de choisir un classifieur capable de prendre en considération une telle caractéristique. L'auteur opte alors pour le classifieur LIBLINEAR [Fan et al., 2008], représentant en réalité une version étendue du classique SVM [Vapnik, 1998], et qui est adaptée aux données de grandes dimensions. En effet, LIBLINEAR est capable de générer un classifieur pour des données pouvant contenir des centaines voire des milliers d'instances et de variables. Pour cela, LIBLINEAR fait appel à la méthode SVM associée au coût L2 (aussi appelé L2-SVM). On renvoie le lecteur pour de plus amples détails aux articles suivants [Pontil and Verri, 1998, Abe, 2002].
L'auteur génère LIBLINEAR sur *Mélanome*, en considérant l'ensemble des données décrivant les images comme étant la matrice des variables explicatives, et le réel label de chaque image comme étant la variable à prédire. De plus, une K-cross validation croisée est effectuée, avec K=20, et les résultats sont présentés en terme de sensibilité et de

spécificité sous la forme d'une courbe ROC. Nous effectuons l'expérimentation de cette étude en générant LIBLINEAR sur l'ensemble du jeu de données *Mélanome*. La courbe ROC obtenue peut être vue sur la Figure 7.3. On remarque que le modèle généré est de bonne qualité, puisque son AUC est de 0.88. Ces résultats sont bien cohérents avec ceux trouvés dans [Wazaefi, 2013].

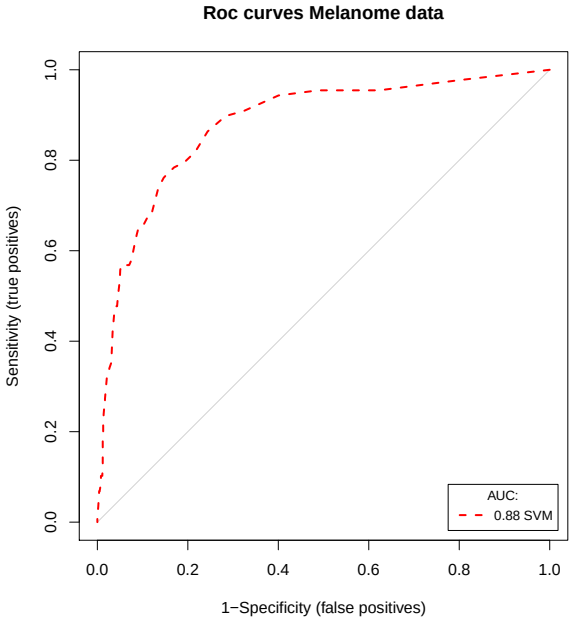


FIGURE 7.3 – Evolution de l'AUC pour LIBLINEAR

Contrairement à cette étude, notre contribution dans ce travail est de générer un classifieur dans un contexte où le réel label de chaque instance est difficile à obtenir, et est alors remplacé par les annotations des différents dermatologues. Les vérités terrains ne seront alors utilisées que dans le but d'évaluer la qualité du classifieur final en comparant le résultat de sa classification avec la vérité terrain.

Compte tenu du jeu de données *Mélanome* récolté, et à partir des trois modèles Ignore, X-Ignore et ExpertS développés dans cette thèse, notre objectif est :

- De produire un classifieur ne prenant en compte que les labels des dermatologues et leur incertitude : on utilise pour cela les deux modèles Ignore et X-Ignore,
- De classer une nouvelle image dans un des deux groupes : nævus malin ou bénin,
- De comparer les résultats obtenus avec d'autres modèles Baseline [Raykar et al., 2010, Yan et al., 2010] ne prenant pas en compte l'incertitude,
- De classer les images d'entrainement en fonction de leur qualité calculée avec le modèle X-Ignore,
- D'estimer la performance de chaque dermatologue et de sélectionner les plus per-

formants, à l'aide du modèle ExpertS,
– D'inférer la qualité d'une nouvelle image.

Avant de procéder à ces analyses, on remarque que *Mélanome* a un nombre de variables largement supérieur au nombre d'images (30720 contre 1097), ce qui est très souvent le cas lors de données médicales. Or cette caractéristique pose souvent des difficultés lorsque l'on souhaite appliquer des méthodes usuelles de classification supervisée. C'est ce que nous allons voir dans la section qui suit.

7.3 Prétraitement des Données de Grande Dimension

7.3.1 Problème lorsque p » n

Depuis une dizaine d'années, les développements en biotechnologie ont permis de collecter une grande quantité de données biologiques (génomiques, protéiniques,...), souvent caractérisées par un petit nombre d'observations ou d'échantillons. Ainsi, il arrive très souvent que les jeux de données aient un nombre de variables de l'ordre de 1000 voire plus, alors que le nombre d'individus est de l'ordre de 10 [West et al., 2001, Dudoit et al., 2002]. Or travailler avec de telles données posent souvent des difficultés en classification. En effet, bien que l'impact de données de grande dimension ne soit pas encore bien compris par l'ensemble des communautés d'apprentissages statistiques, il a été montré dans de nombreux articles que les méthodes habituelles de classification ne sont pas valables en présence de données où le nombre de variables est supérieur au nombre d'instances [Bickel and Levina, 2004, Fan and Lv, 2006]. Par exemple, dans ce contexte, l'estimateur des moindres carrés très souvent nécessaire dans les problèmes de régression ne peut pas être estimé. Des méthodes ont alors été développées pour répondre à ce problème, telles que la méthode PLS (Partial Least Squares) ou encore la régression Ridge. Ces deux méthodes sont détaillées dans [Tenenhaus, 1998]. Elles permettent une réduction de dimension à partir de l'ensemble des variables, ce qui peut être un avantage pour de nombreuses applications. Cependant, dans le cas où le nombre de variables est très nettement supérieur au nombre d'individus, des combinaisons de toutes les variables présentes peuvent très vite devenir ininterprétables. D'autres stratégies ont alors vues le jour, stratégies utilisant la plupart du temps les méthodes "sparses", méthodes permettant la sélection de modèles par pénalisation en produisant un grand nombre de coefficients de variables nuls. On peut citer par exemple les méthodes LASSO, Elastic Nest ou encore S-PLS (Sparse PLS). Nous renvoyons le lecteur à l'ouvrage [Hastie et al., 2009] pour davantage de détails sur ce genre de méthodes.

Les modèles Ignore, X-Ignore et ExpertS n'ayant pas été développés pour répondre au problème de grandes dimensions, il nous a paru important d'effectuer une étape de sélection de variables sur *Mélanome* avant de les expérimenter.

7.3.2 Etape de Sélection de Variables pour Mélanome

Notre objectif dans cette section est d'effectuer une sélection de variables pour le jeu de données *Mélanome*, afin d'arriver à un nombre de variables inférieur au nombre d'individus, et de pouvoir ainsi générer les modèles développés dans cette thèse. Pour cela, de très nombreuses méthodes ont été étudiées [Hoggart et al., 2008, Lê Cao et al., 2011, Lê Cao and Le Gall, 2011]. Nous optons ici pour la méthode S-ACP (Sparse ACP - Analyse en Composante Principale) , qui correspond en réalité à une extension de l'ACP classique et qui permet d'effectuer une sélection de variables grâce à la prise en compte d'une pénalité LASSO [Shen and Huang, 2008]. L'ajout d'une pénalité permet de rechercher des combinaisons comportant un nombre important de coefficients nuls. Nous renvoyons le lecteur à l'ouvrage de [Hastie et al., 2009] pour davantage de détails sur ce genre de méthodes. Des rappels sur l'ACP et sur la Sparse ACP sont présentés en Annexe G .

On décide, une fois la Sparse ACP effectuée sur le jeu de données *Mélanome*, de ne garder que les 10 premières variables significatives du modèle. Ce choix est justifié par le calcul de la variance des nouveaux axes (appelés composantes principales), obtenu par la S-PCA. En effet, l'application de la S-PCA sur un jeu de données conduit à transformer des variables reliées entre elles en de nouvelles variables décorélées (les composantes principales). Or chaque composante principale est associée à une variance, correspondant à son inertie, c'est-à-dire à la mesure de la dispersion totale du nuage de points. Plus la variance est élevée, plus l'axe est significatif. Ainsi, la composante principale 1 de la S-ACP est la composante dont la variance est maximale.
Afin de choisir le nombre de composantes principales que nous allons garder pour le reste de notre étude, nous effectuons une S-PCA sur mélanome, et on calcule la variance associée à chaque composante principale. Les sommes cumulées de la variance à travers les 20 premières composantes principales sont présentées sur la Figure 7.4.

On remarque que la première composante principale explique à elle toute seule plus de 40% de l'inertie du jeu de données. En considérant les 3 premiers axes, on arrive à un taux d'explication de 60%. Nous décidons cependant empiriquement de prendre les 10 premiers axes, afin d'arriver à une variance expliquée proche de 80%. On se retrouve alors finalement avec un jeu de données de dimension 1097×10, nous permettant de générer les modèles Ignore, X-Ignore et ExpertS.

7.4 Expérimentations et Analyse des Résultats

Les dermatologues sont tenus d'annoter chaque image suivant les deux labels bénin ou malin, et d'exprimer leur incertitude quant au label donné en ajoutant à chacun d'eux la mention "certainement" dans le cas où ils sont sûrs, ou "probablement" dans le cas de doute. A partir de ces annotations, il nous ait possible de générer la matrice Y d'annotations des juges, ainsi que la matrice H d'incertitude. Y et H sont générés comme suit. Soit y_i^t (resp.h_i^t) l'annotation (resp. l'incertitude) du dermatologue pour l'image i.

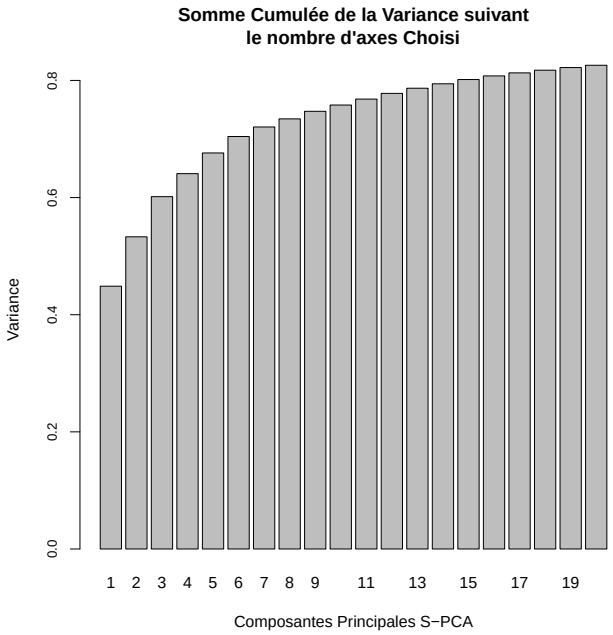


FIGURE 7.4 – Représentation de la Somme Cumulée de la Variance des Composantes Principales obtenue par la S-PCA pour les 20 premières composantes.

Pour chaque dermatologue t et pour chaque image i, si :
- Le nævus est certainement bénin alors $y_i^t = 0$ et l'état d'incertitude $h_i^t = 0$,
- Le nævus est probablement bénin alors $y_i^t = 0$ et l'état d'incertitude $h_i^t = 1$,
- Le nævus est probablement malin alors $y_i^t = 1$ et l'état d'incertitude $h_i^t = 1$,
- Le nævus est certainement malin alors $y_i^t = 1$ et l'état d'incertitude $h_i^t = 0$.

A présent que nous avons défini les matrices Y et H, nous disposons finalement :
- Du jeu de données mélanome X composé de 1097 images et de 10 descripteurs,
- Des véritables labels Z pour chaque image : bénin (noté 0) ou malin (noté 1). On rappelle que 88 images sont dans le groupe 1, le reste étant 0,
- De la matrice des annotations Y des 10 dermatologues : 0 bénin, 1 malin,
- De la matrice d'incertitude H des 10 dermatologues : 0 pour certain, 1 pour probable.

On retrouve bien dans le contexte général des 3 modèles Ignore, X-Ignore et ExpertS. On applique chaque modèle en considérant le protocole K-fold validation croisée, avec K=20. Les résultats obtenus sont présentés et discutés ci-dessous.

7.4.1 Résultats en Présence des 10 Dermatologues Seniors

On compare Ignore et X-Ignore aux deux modèles baselines de référence, à savoir le modèle de Raykar [Raykar et al., 2010] et Yan [Yan et al., 2010], à l'aide de l'estimation de l'AUC. Par ailleurs, la performance de chaque annotateur est estimé en termes de sensibilité et spécificité grâce au modèle Ignore, et ExpertS est ensuite généré afin de sélectionner seulement les meilleurs dermatologues dans la génération du modèle. Enfin, l'utilisation du modèle X-Ignore engendre une estimation de la qualité de chaque image du jeu de données, aboutissant à la possibilité d'inférer la qualité d'une nouvelle image.

Comparaison de Ignore, X-Ignore, Yan et Raykar

Nous utilisons les 4 systèmes Raykar, Yan, Ignore (en considérant les 3 a prioris) et X-Ignore sur le jeu de données *Mélanome*, et nous estimons en moyenne l'AUC correspondant pour chacun des modèles. Les résultats obtenus peuvent être vus sur la Figure 7.5. On remarque que tous les modèles ont environ le même AUC, tous égaux à 0.84 ou 0.85 suivant le classifieur. Ce résultat peut à première vue sembler étrange, puisque nos modèles prenant en compte l'incertitude ne donnent pas de meilleurs résultats que les modèles baselines de référence. Néanmoins, ce résultat est expliqué par le fait que les dix dermatologues sont séniors ; ils ont donc de fortes chances d'être tous a priori experts et d'avoir ainsi un niveau d'erreurs sur les labels très faible. Or, nous avons montré dans les chapitres 4 et 5 que les modèles Ignore et X-Ignore sont significativement meilleurs que les modèles baselines dans un contexte de classification supervisée en présence de multiples annotateurs naïfs, c'est-à-dire en présence d'annotateurs qui ont un niveau de connaissance très hétérogènes. Nous reviendrons à ce problème plus tard dans le chapitre. On estime dans la section qui suit la performance des annotateurs, et on montre que ces derniers sont en réalité tous experts dans le domaine.

Performance des Annotateurs

Le modèle Ignore permet d'obtenir une estimation de la performance des 10 dermatologues en terme de sensibilité et de spécificité dans des situations de connaissance et d'incertitude. Leur performance est ainsi estimée dans ces deux situations séparément. La Figure 7.6 montre les résultats obtenus. Nous reprenons les termes utilisés dans [Raykar and Yu, 2012] pour qualifier 4 groupes d'annotateurs : les spammers et les experts étant déjà définis, les annotateurs biaisés représentent ceux qui annotent toujours les instances dans le même groupe, et les annotateurs malicieux représentent les annotateurs qui répondent très souvent le contraire du label réel.

D'après la Figure 7.6, on note que dans les situations de connaissance tous les dermatologues sont bien experts dans le domaine, puisqu'ils ont tous une sensibilité comprise entre 0.75 et 0.90, et une spécificité comprise entre 0.90 et 1. Dans les situations incertaines, ces derniers ont tout de même tous un taux de vrai positifs et de vrais négatifs supérieur à 0.60, ce qui est tout de même élevé. Par ailleurs, 8 dermatologues sur 10

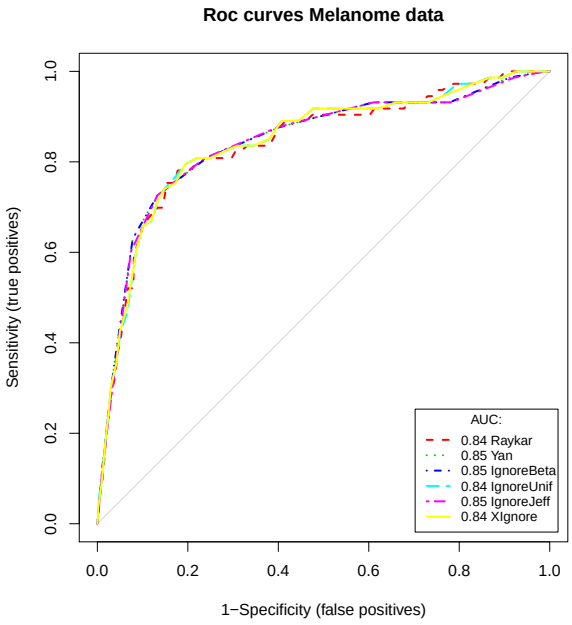


FIGURE 7.5 – Evolution de l'AUC : comparaison entre 2 Modèles incluant l'incertitude des annotateurs (Ignore et X-Ignore) et 2 modèles Baselines (Yan et Raykar).

ont un taux de vrais négatifs (spécificité) supérieur à 0.75 dans les situations de doute. Ainsi, les annotateurs classent en général mieux les images du groupe 0 que les images du groupe 1. Cette observation peut en fait être expliquée par le fait que *Mélanome* est composée de 88 images dans la classe malin (classe 1), le reste étant dans la classe bénin (classe 0) : en présence de classes déséquilibrées, le classifieur à tendance à biaiser les résultats en faveur de la classe dominante, correspondant à la classe bénin dans notre cas.
Le problème de classes déséquilibrées est un problème très étudié de nos jours en apprentissage. Il ne nous a cependant pas semblé important dans cette étude de l'approfondir, notre objectif n'étant pas de générer le meilleur classifieur possible, mais de comparer les résultats de nos approches avec les résultats d'approches précédemment développées pour un même jeu de données. Le lecteur peut tout de même se référer aux articles [Ting, 2000, Elkan, 2001] pour de plus amples informations à ce sujet.

Sélection des Annotateurs Experts

Nous venons de montrer que tous les dermatologues ayant annoté les données peuvent être qualifiés d'experts. Nous décidons néanmoins d'utiliser le système ExpertS dans le but de voir s'il ne serait tout de même pas possible d'améliorer la performance du classifieur en éliminant les dermatologues les moins performants parmi eux. ExpertS est alors utilisé sur l'ensemble des données, et on l'évalue toujours à l'aide d'un K-cross validation

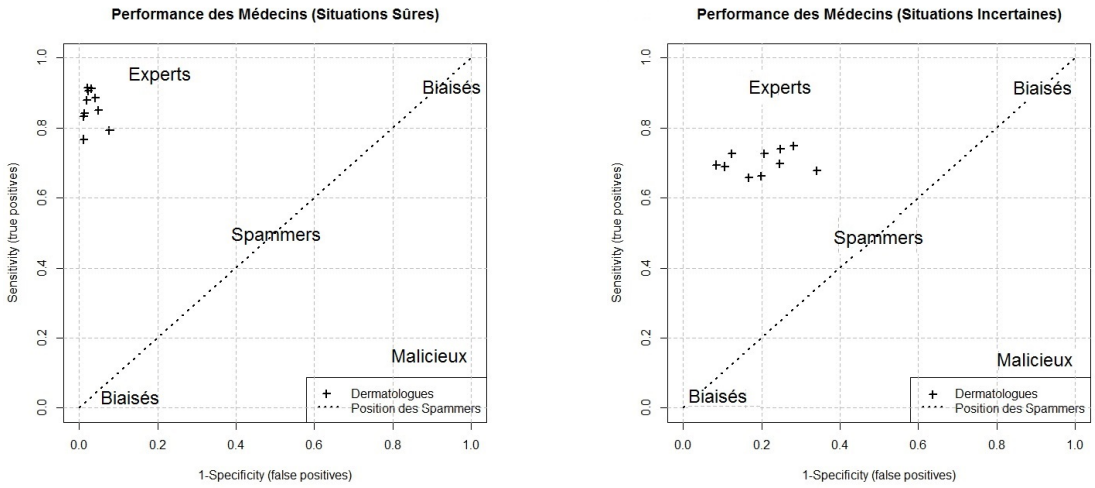


FIGURE 7.6 – Représentation de la Performance des Annotateurs en Terme de Sensitivité et de Spécificité dans les Situations Certaines (à gauche) et Incertaines (à droite).

croisée (K=20). La moyenne des AUC obtenus est représentée sur la Figure 7.7, est a une valeur de 0.86 (contre 0.84 ou 0.85 pour Ignore et X-Ignore). Le modèle généré s'est donc très légèrement amélioré, ce qui semble être un résultat logique au vu des performances élevées de tous les dermatologues.
Par ailleurs, à la fin de chaque exécution de ExpertS, on obtient le classement des dermatologues par ordre de performance. Or l'algorithme a été exécuté plus de 50 fois puisque l'on a effectué une K-cross validation croisée avec K=20. On calcule alors le nombre d'occurrences de chaque annotateur dans chaque classement. La Figure 7.8 montre les résultats obtenus. La classe en noire sur l'histogramme représente le réel classement de chaque dermatologue, obtenu en comparant les annotations de chaque dermatologue par rapport aux vérités terrains (le dermatologue le plus performant (resp. le moins performant) étant celui qui a le plus grand nombre (resp. le plus petit nombre) de labels exacts).
On note que le classement obtenu par ExpertS (autrement dit la classe ayant obtenue le plus grand nombre d'occurence) est en accord avec le réel classement pour 5 dermatologues sur 10 (les dermatologues 2, 4, 5, 7 et 10). Concernant les autres dermatologues, on remarque qu'ExpertS leur attribue une classe très proche de leur réel classement. En effet, nous pouvons prendre l'exemple du dermatologue 1 où ExpertS l'a classé le plus souvent en 4ème position, alors qu'il est en réalité en 5ème position par rapport aux autres dermatologues. De même, le dermatologue 8 a été le plus souvent classé en 7ème position, alors que sa réelle classe est 8. On montre ainsi la réelle faculté de l'algorithme

ExpertS a sélectionné parmi l'ensemble des annotateurs les plus performants d'entre eux.

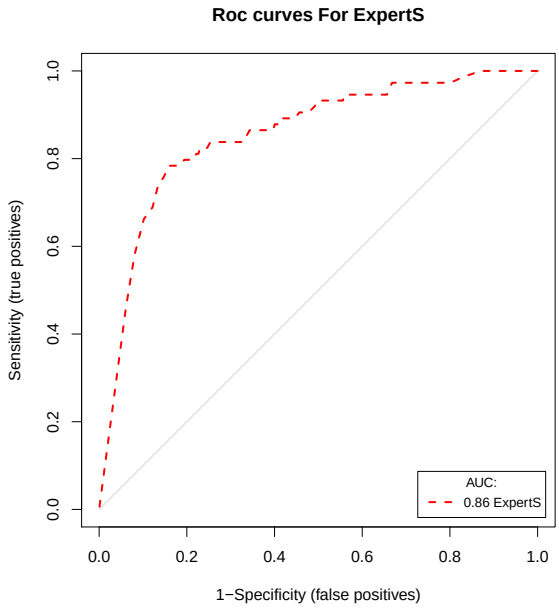


FIGURE 7.7 – Représentation de l'AUC pour ExpertS avec les 3 Annotateurs Sélectionnés.

Qualité des Données

Jusqu'à présent nous avons estimé, grâce au modèle Ignore et ExpertS, la performance de chaque dermatologue, puis nous avons sélectionné les meilleurs d'entre eux pour la génération du classifieur. Il est cependant possible d'aller plus loin dans l'étude avec l'utilisation du modèle X-Ignore, puisque ce dernier estime en plus la qualité de chaque image du jeu de données. Une image peut être nommée de bonne qualité dans le cas où elle est claire, nette, et dans le cas où ses caractéristiques permettent de l'attribuer à une des deux classes assez facilement. Dans le cas d'une image de mauvaise qualité, cette dernière a de très fortes chances d'induire les annotateurs a l'erreur, malgrè le fait qu'ils soient performants.
Nous avons expérimenté X-Ignore sur la totalité du jeu de données (1097 images). Les résultats de l'estimation de la qualité de chaque image peuvent être vus à la Figure 7.9. On remarque que deux groupes d'instances se distinguent très nettement ; les images dont l'estimation de la qualité se situe dans l'intervalle [0.2,0.4], et les images dont la qualité se trouve dans l'intervalle [0.45,0.65]. On peut en conclure que le premier groupe d'images représente les images de moins bonnes qualités (et qui seraient alors au nombre de 442), alors que le deuxième groupe constituerait les images de meilleure qualité (655 images). Cette division en deux groupes peut être expliquée par le choix de l'a priori

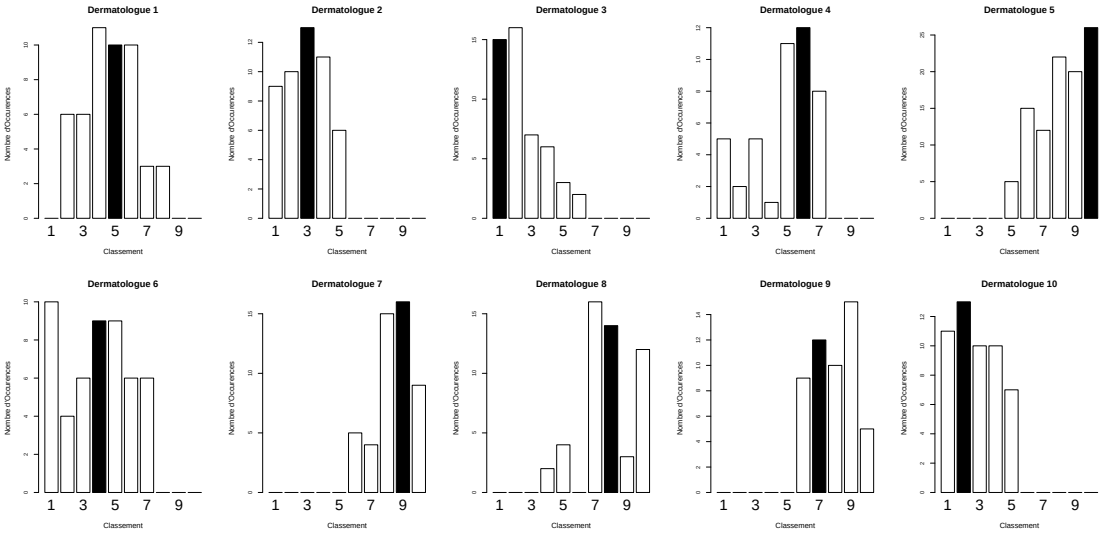


FIGURE 7.8 – Histogramme du Nombre d'Occurences de Chaque Dermatologue pour Chaque Classe avec l'utilisation d'ExpertS. La classe en noire représente leur réel classement (obtenu en comparant les annotations de chaque dermatologue par rapport aux vérités terrains).

que nous avions fait sur les images. En effet, dans la section 5.1.4, nous avions supposé un a priori Gaussien de moyenne -2 pour les instances dont plus de la moitié des annotateurs étaient incertains, et de 2 pour les autres. Or cette hypothèse se trouve ici être un peu forte, puisqu'elle aboutit à une séparation binaire des images : celles possédant un nombre de labels certains supérieur ou égale à 5, contre celles possédant moins de 5 labels certains. Pour cette raison, nous décidons de faire appel à un autre a priori, dépendant cette fois ci du nombre exacte d'images annotées par les dermatologues de façon sûres et incertaines. Ainsi, on utilise toujours un a priori Gaussien, mais la moyenne μ est cette fois calculée par la fonction $y = 4x - 2$, où x représente le taux de labels où les annotateurs sont certains ($x \in [0, 1]$). Ce taux est calculé pour chaque image. De cette manière, plus le taux de labels certains est proche de 1, plus la moyenne de l'a priori gaussien est élevé et (proche de 2). A l'opposé, plus le taux est proche de 0, plus l'a priori est faible et donc proche de la valeur -2. Les résultats de l'estimation de la qualité des données avec ce nouveau a priori peuvent être vus sur la Figure 7.10. Dans ce contexte, on remarque que la distribution de la qualité des images suit approximativement une loi Gaussienne, avec un grand nombre d'images de qualité moyenne regroupé dans un même groupe (258 images), tandis que les images de moins bonnes qualités sont au nombre de 449, les images de meilleurs qualités sont au nombre de 390. Ainsi, plus une image a une qualité proche de 0.225, plus elle sera de meilleure qualité.

Nous avons jusqu'à présent supposé qu'une image était de plus ou moins bonne qualité suivant le nombre d'annotateurs incertains de leur labels. Cet indice peut cependant

être discuté. En effet, il est possible, par exemple, d'opter pour d'autres indices, tels que compter le nombre de dermatologues en accord sur le label de l'image. On suppose alors dans ce cas de figure que plus une image est de bonne qualité, plus elle aura un nombre de labels équivalents élevé, puisque les dermatologues auront plus de chances d'être d'accord entre eux.
Ainsi, on suppose à présent cet indice, et on compte pour chaque instance le nombre de labels équivalents. L'histogramme présenté sur la Figure 7.11 montre les résultats obtenus. Le 1er groupe des instances, noté "0 DIFF" sur le graphique, contient toutes les instances pour lesquelles il y a consensus (tous les dermatologues sont d'accord entre eux). Concernant le 2ème groupe noté "1 DIFF", il contient toutes les instances dont 1 seul annotateur est en contradiction avec les autres. En d'autres termes, un annotateur a annoté 1 (resp. 0) alors que tous les autres ont annoté 0 (resp.1). Et ainsi de suite pour les autres groupes. On remarque que les deux premiers groupes contiennent à eux deux 700 images sur les 1097. Ceci montre qu'une grande partie des images présentes dans le jeu de données est de bonne qualité.

Le but à présent est de comparer les résultats trouvés avec ce dernier indice (basé sur le consensus) et les résultats trouvés avec l'indice précédent (basé sur le nombre d'annotateurs incertains), afin de voir s'il existe une possible relation entre les deux. En effet, il est logique de penser qu'une instance de très bonne qualité soit une instance où la plupart des dermatologues sont d'accord entre eux (2ème indice) et où la plupart sont, par ailleurs, certains du label donné (1er indice).
La Figure 7.12 représente la qualité des images pour chacun des groupes obtenue sur l'histogramme 7.11. Ce graphique montre que la plupart des images de bonnes qualités trouvées avec le 1er indice (cf Graphique Gauche 7.10) sont présents dans les deux premiers groupes constituant les images consensuelles, c'est-à-dire dans les groupes où les annotateurs sont le plus d'accord entre eux. Il est d'ailleurs à noter que parmi les 29 meilleures instances de l'indice utilisé dans X-Ignore (voir Graphique 7.10), 26 sont retrouvées dans le 1er groupe en utilisant l'indice du consensus, tandis que les 3 restantes sont dans le 2ème. A l'opposé, les images de moins bonnes qualités (avec un indice inférieur à 0.3) avec l'indice X-Ignore, sont pour la majeur partie dans les groupes 3, 4 ou 5 avec le consensus.
Il existe donc bien une relation entre la qualité de l'image, le nombre de labels équivalents donnés par les annotateurs (2nd indice), ainsi que l'incertitude des annotateurs (1er indice). En effet, plus l'image est de bonne qualité, plus les annotateurs sont d'accord entre eux, et plus leur taux d'incertitude diminue.

Prédire la Qualité des Données

Nous avons à présent, pour chaque image du jeu de données, pu estimer sa qualité. Il serait intéressant de voir s'il serait possible de prédire la qualité d'une nouvelle image. En effet, chaque image est décrite par les 10 variables obtenues par la S-PCA, et est associée à une valeur réelle correspondant à sa qualité. Nous décidons d'effectuer alors une régression linéaire sur l'ensemble de ce jeu de données, afin d'obtenir un modèle qui

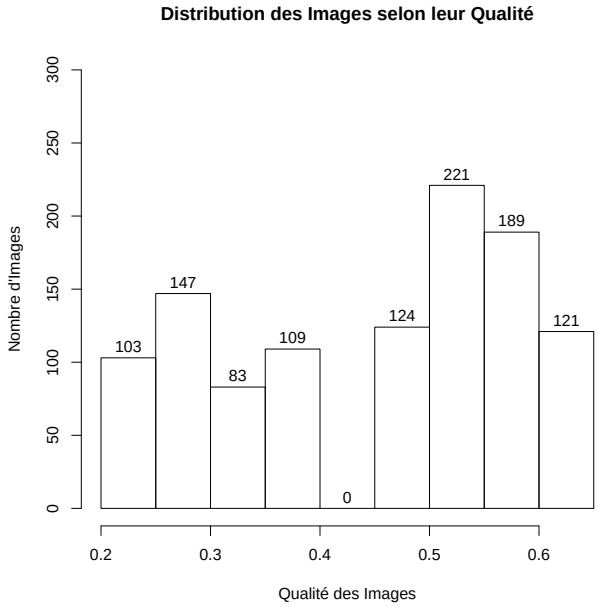


FIGURE 7.9 – Estimation de la Qualité de Chaque Image du Jeu de Données Mélanome à l'aide du Modèle X-Ignore : Représentation des résultats par un histogramme regroupant les instances selon leur qualité.

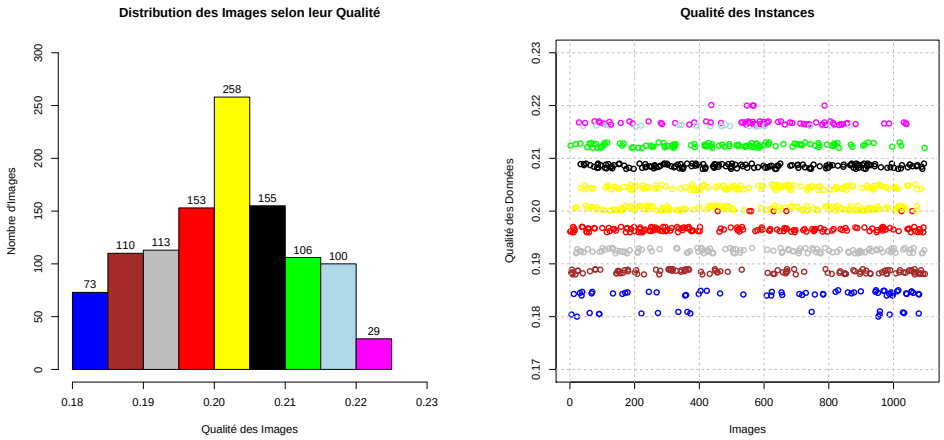


FIGURE 7.10 – Représentation de la Qualité des Instances à l'aide du Modèle X-Ignore : (à Gauche) Représentation des résultats par un histogramme regroupant les instances selon leur qualité. (à Droite) Représentation des instances selon l'estimation de leur qualité. Les couleurs correspondent aux différents groupes de l'histogramme de droite. Les couleurs ont pour but de montrer les images équivalentes entre le graphique de droite et celui de gauche.

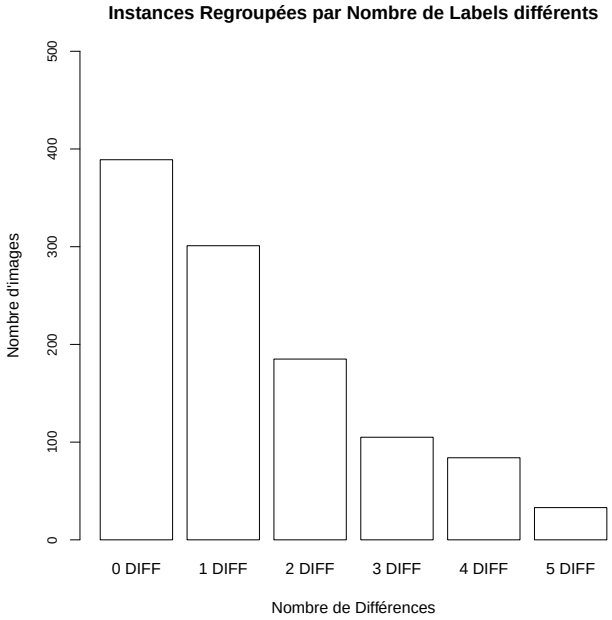


FIGURE 7.11 – Regroupement des Images en fonction du Nombre de Labels Equivalents.

prédit la qualité d'une nouvelle instance. Une K-cross validation croisée a été effectuée, toujours avec K=20. Les résultats peuvent être vus sur le Graphique 7.13. Le graphique de haut représente chaque image selon la valeur de sa qualité prédite par la régression Val_{Reg}, et la valeur de sa qualité prédite par le modèle XIgnore Val_{XIgn}. Un modèle de régression idéal serait alors que toutes les images se trouvent sur la droite d'équation $Val_{Reg} = Val_{XIgn}$, et plus la différence entre ces deux valeurs est petite, meilleur est le modèle de prédiction. Ainsi, pour mieux visualiser les résultats obtenus, on calcule l'erreur de prédiction pour chaque image. On pose ϵ_i l'erreur de prédiction pour l'image i. On a alors :

$$\epsilon_i = (Val_{Xign} - Val_{Reg})^2$$

Les images sont ensuite regroupées en histogramme, selon la valeur de l'erreur trouvée. On remarque que plus de la moitié des images ont un taux d'erreur de prédiction inférieur à 5×10^{-5} et plus de 80% ont un taux d'erreurs inférieur 2×10^{-4}. La somme du taux d'erreurs de prédiction totale est de 9×10^{-2}. Ceci montre la bonne qualité de prédiction du modèle.

Les résultats des expérimentations de cette première partie nous ont permis de constater la haute performance de tous les dermatologues séniors. Dans un tel contexte d'annotations, les modèles Ignore et X-Ignore développés dans cette thèse sont aussi performants que les modèles de la littérature. Ce résultat est non négligeable, puisqu'il rassure sur la stabilité et la performance de nos modèles comparées aux modèles base-

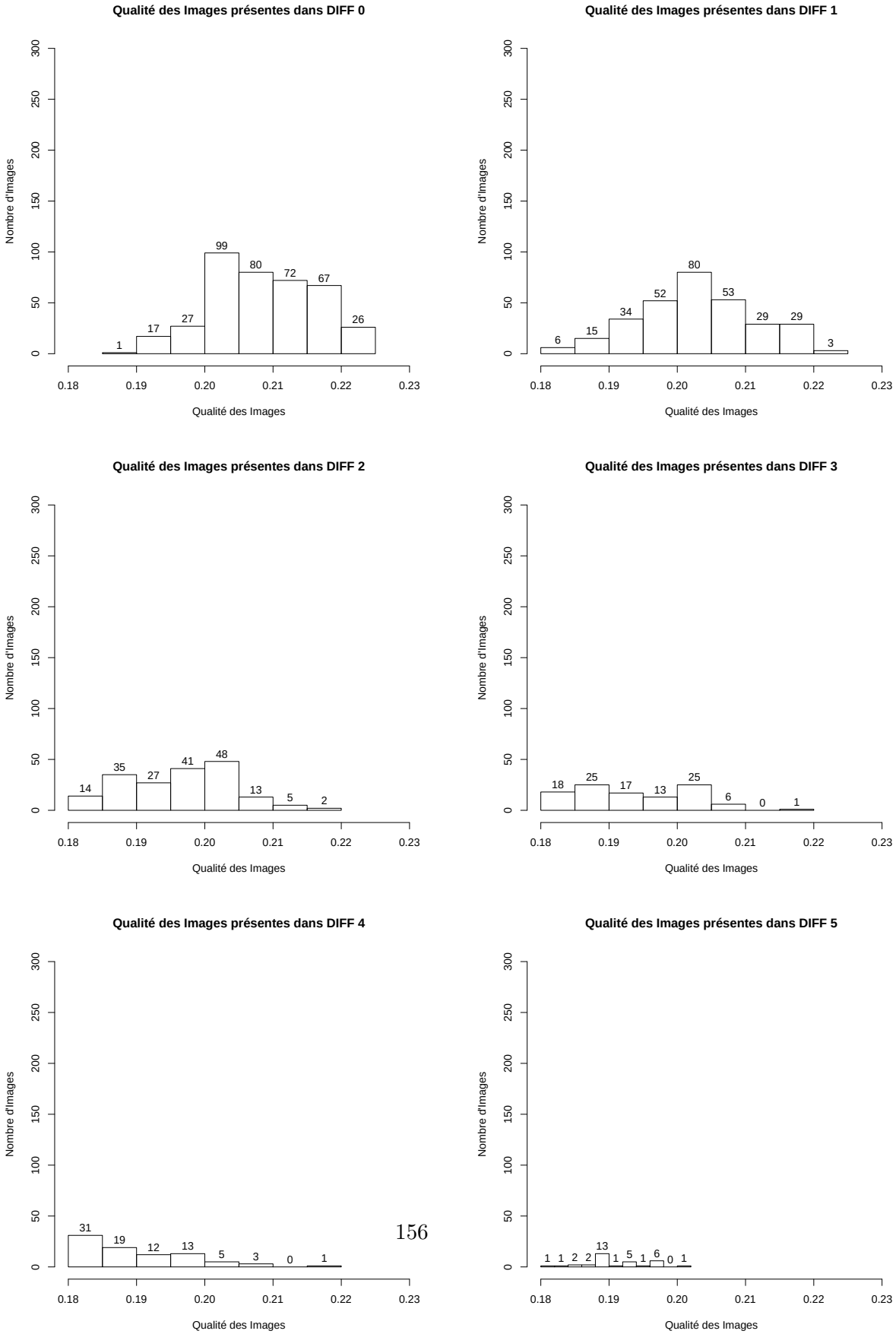


FIGURE 7.12 – Représentation de la Qualité des Images pour chaque Groupe d'Images ayant un même Nombre de Labels Similaires.

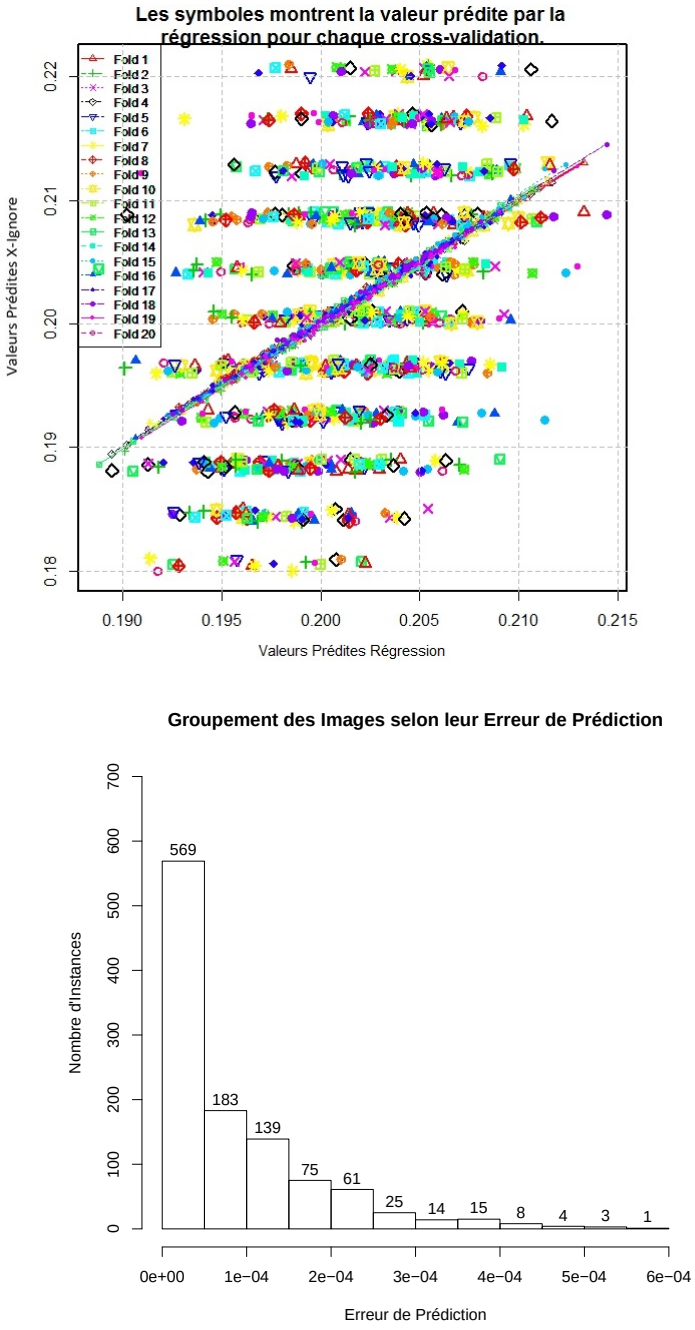


FIGURE 7.13 – Résultats du modèle de Régression Linéaire pour la Prédiction de la Qualité d'une Nouvelle Image. (en Haut) L'axe des abscisses correspond à la valeur estimé de la qualité de l'instance par la régression linéaire, et l'axe des ordonnées correspond à la valeur initiale obtenue par le modèle X-Ignore. (en Bas) Calcul de la différence entre la valeur estimée par la régression et la valeur X-Ignore de départ, et représentation des résultats sous la forme d'un histogramme regroupant les instances selon la valeur de cette différence.

lines de référence, en présence d'annotateurs compétents.
Nous avons cependant décidé d'aller plus loin dans notre étude en simulant 30 juges non experts ou spammers et en regénérant les classifieurs en présence d'annotateurs experts (les dermatologues) et non experts. Les résultats obtenus sont étudiés et discutés dans la section qui suit.

7.4.2 Ajout de 30 Dermatologues Non Experts

Simulation des Annotateurs

Le temps pour annoter manuellement nos données par des personnes non expertes est très long, et nous n'avons pas pu avoir de telles annotations avant la fin de la thèse. On décide alors de simuler 30 annotateurs non experts de la façon suivante : nous divisons aléatoirement le jeu de données mélanome en deux groupes Mel_{doute} et Mel_{sur}, représentant respectivement 60% et 40% de melanome. On suppose que les annotateurs ont entre 60 et 90% d'incertitude sur les images de Mel_{doute}, et ont entre 30 et 50% d'erreurs pour les instances appartenant à Mel_{sur}.

Comparaison de Ignore, X-Ignore, Yan et Raykar

Nous expérimentons de nouveau tous les modèles Ignore, X-Ignore, Yan et Raykar pour le jeu de données *Mélanome*, mais cette fois en prenant en compte 40 annotateurs, les 10 dermatologues experts et 30 annotateurs non experts simulés. Les résultats des AUC estimés peuvent être vus sur la Figure 7.14. Dans un tel contexte, l'AUC pour le modèle de Raykar chute à 0.80 (alors qu'il était de 0.84 lorsque l'on considérait uniquement les dermatologues experts), alors que l'AUC pour les modèles Ignore et X-Ignore proposés reste beaucoup plus stable puisqu'il reste aux alentours de 0.83, 0.84 ou 0.85 pour les trois a prioris. Ainsi, les résultats obtenus ici confirment bien l'avantage des modèles Ignore et X-Ignore proposés, comparé aux modèles baselines de référence, dans un contexte où de nombreux juges ne sont pas compétents. La prochaine section étudie la performance de chaque annotateur.

Performance des Annotateurs

Contrairement à X-Ignore, le modèle Ignore permet d'estimer la performance de chaque annotateur en terme de sensibilité et de spécificité pour les situations de connaissance et d'incertitude. Nous utilisons alors Ignore pour représenter la performance de chaque annotateur. Les résultats peuvent être vus à la Figure 7.15. On remarque que les deux groupes d'annotateurs sont bien identifiables sur les graphiques, où la performance des dermatologues experts est nettement supérieure à celles de spammers, que ce soit dans les situations sûres ou non. Ce résultat semble tout à fait en accord avec les données initiales, puisque l'on a vu dans la section précédente que les dermatologues experts avaient une performance élevée dans chacune des deux situations. Or les spammers ont été simulés avec un taux d'erreurs de classification plus important.

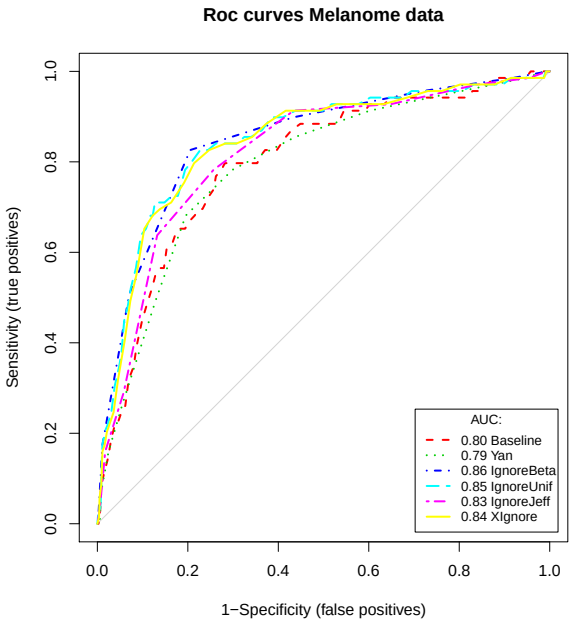


FIGURE 7.14 – Evolution de l'AUC avec Ajout de 30 Annotateurs Spammers : Comparaison entre 2 Modèles incluant l'incertitude des annotateurs (Ignore et X-Ignore) et 2 modèles Baselines (Yan et Raykar).

Sélection des Annotateurs Experts

On reprend maintenant le modèle ExpertS, afin de vérifier qu'il sélectionne bien les dermatologues séniors et que le classifieur augmente en performance. Concernant la sélection des annotateurs, ExpertS élimine bien tous les annotateurs spammers, puisqu'il ne génère le classifieur qu'en prenant en considération les 10 dermatologues séniors du jeu de données initial. De plus, on retrouve bien la valeur de l'AUC estimé avant la simulation des 30 annotateurs non experts. Ce résultat peut être vu sur la Figure 7.16.

7.5 Conclusion

Nous avons évalué dans ce chapitre chacune des contributions apportée dans cette thèse, à savoir les modèles Ignore, X-Ignore et ExpertS, sur une application médicale réelle, mélanome. Cette étude nous a permis de montrer une nouvelle fois la performance de nos modèles dans un contexte de classification supervisée en présence de multiples annotateurs non obligatoirement experts. Dans le cas où les annotateurs sont tous experts, nos modèles sont aussi performants que les précédents modèles Baselines développés. Cependant, le réel intérêt des modèles développés ici peut être observé dans le cas où de nombreux annotateurs sont spammers, où nos modèles sont très clairement plus stables dans de telles conditions. De plus, la génération de ExpertS sur le jeu de

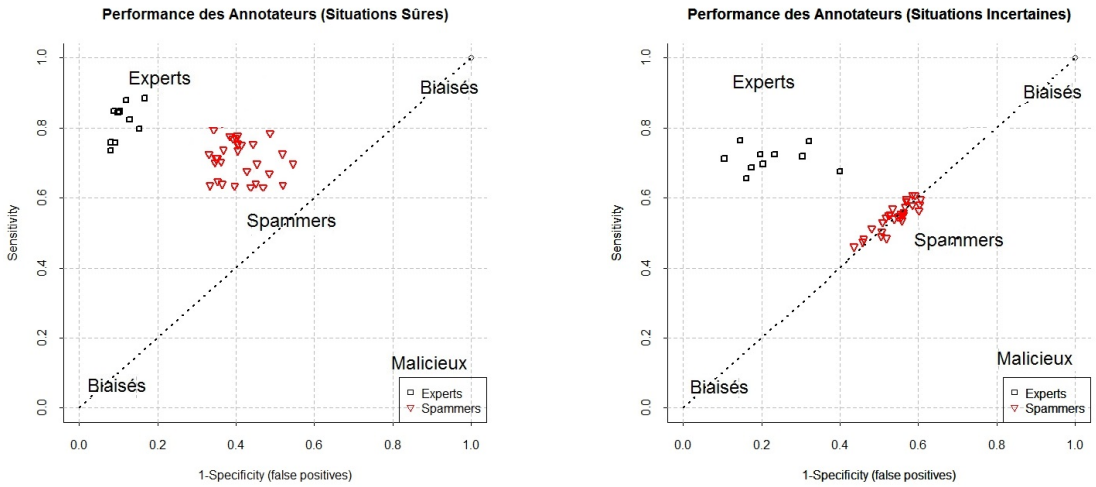


FIGURE 7.15 – Représentation de la Performance des 40 Annotateurs en Terme de Sensitivité et de Spécificité dans les Situations Certaines (à gauche) et Incertaines (à droite).

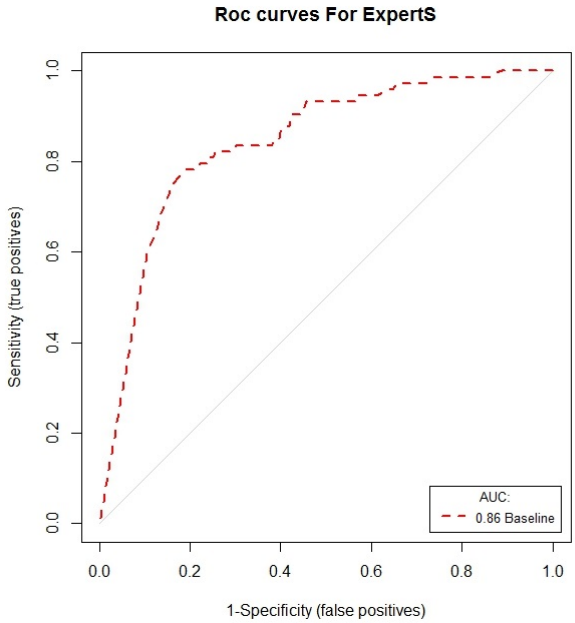


FIGURE 7.16 – Ajout de 30 Annotateurs Non Experts : Représentation de l'AUC pour le modèle ExpertS, une fois les annotateurs experts sélectionnés.

données mélanome en présence de multiples dermatologues non experts ou spammers a montré l'efficacité de cet algorithme dans la sélection et le classement des annotateurs. Enfin, l'avantage supplémentaire de X-Ignore est qu'il nous permet d'estimer et regrouper les instances selon leur qualité. Les modèles pourraient alors être regénérés en prenant seulement en considération les images de meilleure qualité.

Chapitre 8

Conclusion et Perspectives

Dans le contexte de données annotés par de multiples juges à la place du réel label, le problème de la génération d'un classifieur se pose pour de multiples chercheurs de la communauté statistiques d'aujourd'hui. En effet, l'adaptation des méthodes usuelles de classification supervisée n'est pas si évidente qu'elle n'y parait : le véritable label est remplacé par de multiples labels bruités, les annotateurs pouvant avoir chacun un niveau d'expertise différent. Par ailleurs, avec l'évolution d'infrastructures tels que Internet, de nombreux services ont vu le jour, favorisant les étapes d'annotations à grande échelle par des centaines voir des milliers d'utilisateurs anonymes. Ce procédé est également connu sous le nom anglais de *crowdsourcing*, et engendre un nouveau problème de traitement de données de grande dimension.
Une première étape consistait alors à générer dans ce contexte un classifieur intégrant simultanément l'incertitude des annotateurs ainsi que leurs labels. Lors de cette approche, un choix d'a priori a dû être effectué, afin de prendre en compte les différents niveaux de connaissance des juges. D'autres méthodes ont été précédement développés pour générer un classifieur dans un contexte de classification en présence d'annotateurs multiples, mais aucune ne prenait en considération l'incertitude de ces derniers. Par ailleurs, nous avons dans un deuxième temps affiné le modèle en supposant que l'étape d'annotations des juges dépendait non seulement de leur niveau de connaissance, mais aussi de la qualité des données présentées. Ceci a ainsi fait l'objet du développement de deux nouvelles méthodes de classification, dans le cas de données annotés par de multiples juges à la place du réel label. Pour finir, pour répondre au problème de données de grande dimension, nous avons développé un troisième algorithme ne générant le classifieur qu'à la suite d'une étape de sélection des annotateurs. Nous avons montré l'efficacité de la méthode développée, et l'avantage de notre méthode en terme de temps d'exécution comparée à d'autres algorithmes de sélection .

8.1 Apport de la Thèse

Les deux premières méthodes développées génère un classifieur dans un contexte de classification supervisée en présence de multiples annotateurs non experts. La première

méthode, appelée Ignore, est une extension du modèle baseline de [Raykar et al., 2010] intégrant, en plus des annotations des juges, l'incertitude de ces derniers. A la différence de [Raykar et al., 2010], chaque label donné est associé à un taux d'incertitude qui sera alors exprimé en terme d'a priori lors de la génération du classifieur. Le second modèle, appelé X-Ignore, est une extension du modèle Ignore, incluant en plus de l'incertitude des juges, la qualité des données. En effet, nous supposons dans ce modèle que la performance de chaque annotateur dépend non seulement de son niveau de connaissance, mais aussi de la qualité des données présentées. Cette nouveauté permet l'intégration de nouveaux paramètres estimant la qualité de chaque instance du jeu de données.
Ignore et X-Ignore représentent ainsi deux nouvelles approches bayésiennes probabilistes, répondant au problème de classification en présence de multiples annotateurs non experts. Ces deux approches ont tout d'abord été développées dans un cadre de classification binaire avec incertitude totale, où les juges ont seulement la possibilité d'exprimer leur certitude par les deux choix : certain/incertain, sans pouvoir associé un taux à ce dernier. Cependant, nous avons étendu ces deux modèles à des contextes d'étude beaucoup plus généraux : le contexte multiclasses et le cas de l'incertitude partielle, ou chaque image est ici associé à un degré d'incertitude.
Ces deux modèles ont été testés pour de multiples jeux de données de l'UCI Machine Learning Repository, et ont été comparés à plusieurs modèles baselines n'incluant pas l'incertitude des juges. Il a alors été montré que l'intégration de cette dernière est un critère important pour la stabilité et la performance du classifieur, surtout en présence de multiples annotateurs non compétents ; en effet, alors que les modèles baselines de référence s'effondrent en présence de multiples annotateurs non compétents, les deux méthodes développées ici restent beaucoup plus stables et robustes. Or de nos jours, avec l'utilisation des services de crowdsourcing via Internet, l'étape d'annotation des jeux de données se fait de plus en plus souvent en ligne, en présence de milliers d'annotateurs anonymes et non experts. Pour cette raison, l'utilisation d'Ignore et X-Ignore s'avèrent être deux modèles très prometteurs, prennant en considération les différentes difficultés que présentent les jeux de données actuels.

Le troisième modèle nommé ExpertS développé dans cette thèse considère un tout autre problème : la sélection des experts parmi les annotateurs pour de la génération du classifieur. Avec l'utilisation des services de crowdsourcing, les jeux de données deviennent de plus en plus volumineux puisqu'il est devient très facile de collecter des centaines voir des milliers d'annotations en très peu de temps. Or l'utilisation de ces services web présente le principal inconvénient de ne pas pouvoir avoir de contrôle sur la performance des individus participant à l'étape d'annotation du jeu de données, entrainant alors la possibilité d'obtenir un grand nombre d'annotateurs incompétents et de labels bruités. Or les méthodes précédements décrites n'effectuent pas de sélection d'annotateurs, ce qui peut entrainer une baisse de la qualité du classifieur généré et un temps de calcul relativement long en présence de gros volumes de données. Afin de répondre à ce problème, nous avons développé le modèle ExpertS, modèle à nouveau basé sur l'algorithme baseline de [Raykar et al., 2010], mais intégrant la combinaison de deux

métriques, le spammer score et l'entropie, afin de sélectionner les annotateurs les plus compétents pour la génération du classifieur. Les différentes expérimentations effectuées ont montré l'efficacité de ExpertS dans la sélection des annotateurs, ainsi que l'amélioration de la performance du classifieur généré comparé aux autres modèles n'effectuant pas d'étapes de sélection. De plus, ExpertS a l'avantage par la combinaison des deux scores, d'aboutir à une rapidité en temps d'exécution de l'algorithme, même en présence de gros volumes de données.

Les trois contributions Ignore, X-Ignore et ExpertS ont finalement été testées et validées sur une application médicale réelle, consistant en la reconnaissance de mélanomes à partir d'images annotées par de multiples dermatologues.

8.2 Limitations Rencontrées

Au cours de ce travail de recherche, nous avons été confrontés à certaines limitations qui ouvrent de nouvelles perspectives de recherches. Une de ces limites concerne tous les choix des paramètres d'initialisation (lors de l'algorithme EM entre autre) ainsi que tous les a prioris choisis sur les paramètres des modèles à estimer. En effet, concernant les paramètres d'initialisation de l'algorithme EM, le choix des valeurs initiales des paramètres a plutôt été effectué suivant les précédentes études qui ont été publiées dans ce domaine, à savoir les algorithmes de référence [Raykar et al., 2010, Yan et al., 2010, Raykar and Yu, 2012]. Mais ce choix peut être discuté et d'autres paramètres d'initialisation peuvent être essayés, avec peut être la possibilité d'obtenir de cette façon de meilleurs classifieurs. En ce qui concerne la modélisation de l'incertitude, nous avons opté dans cette thèse pour une approche bayésienne, car cette approche nous semblait la plus facile à comprendre, à interpréter et à modéliser. Cependant, cette modélisation peut aussi être discutée, puisqu'elle nécessite de fixer des distributions a prioris sur chacun des paramètres à estimer, distributions qui sont le plus souvent inconnues et que l'on cherche à estimer et à approcher. Enfin, dans notre travail de recherche, nous avons fait appel à l'algorithme LBFGS et la méthode quasi-newton de façon itérative lors de la maximisation de la vraisemblance. Or les calculs en itérant ces méthodes étaient relativement lourds et complexes. L'algorithme nécessite alors beaucoup de temps de calcul. Il serait peut être intéressant de voir comment il serait possible de faire face à ce problème, en étudiant d'autres méthodes (par exemple méthode de Monte Carlo couplées à des modèles de Markov).

Tout au long de ce travail, nous avons supposé que les juges annotent toutes les données, qu'ils soient ignorants ou incertains. Or cette hypothèse peut paraitre dans certains cas difficile à réaliser, puisque de nos jours il existe de plus en plus de données de très grandes dimensions, composées de milliers d'instances. Dans un tel contexte, il est impossible que chaque annotateur voit toutes les instances, ce qui amène à des matrices d'annotations creuses. Une méthode pour répondre à ce problème serait par exemple de regrouper tous les annotateurs ayant annotés les mêmes instances ensemble, puis de

générer les différents modèles sur chaque groupe d'annotateurs. Enfin, le réel label pour une instance serait obtenu en identifiant le label qui a été le plus souvent répété parmi tous les classifieurs.

Une autre limite de cette thèse concerne le modèle ExpertS développé dans la troisième partie. Ce modèle n'est utilisable que sur des données binaires, l'entropie choisie pour la sélection des annotateurs n'étant pas adaptable aux données multiclasses. Il serait alors intéressant de voir s'il est possible de trouver une autre entropie qui serait cette fois adaptée aux données binaires ainsi qu'aux données multiclasses.

Enfin, lors des expérimentations sur le jeu de données réel mélanome, nous avons été constraint d'effectuer une pré-sélection des variables à l'aide de la Sparse ACP. Nous avions alors sélectionné les 10 premières variables significatives. Or ce choix peut sembler trop sélectif ou trop brusque et peut être discuté, puisque le jeu de données contenait initialement plus de 30000 variables.

8.3 Perspectives

De nos jours, en apprentissage supervisé le problème de la classification en présence de multiples annotateurs biaisés est un sujet d'actualité et de nombreuses problèmatiques restent encore ouvertes dans ce domaine. Un des premiers problèmes qui n'a pas été étudié dans cette thèse est l'extension des modèles Ignore, X-Ignore et ExperTs aux cas de classification multilabels. En effet, dès le début de cette thèse, nous nous sommes posés dans un cadre de classification où chaque instance ne peut être rattachée qu'à une seule classe. Or on constate qu'aujourd'hui, de très nombreuses applications réelles font appel à la classification multilabels ; par exemple dans le domaine de la biologie avec la recherche des différentes fonctions d'une proteine, ou encore dans la classification d'images et de textes : une proteine peut en effet avoir plusieurs fonctions, et une image peut appartenir à plusieurs classes. Ainsi, dans un contexte de classification multilabels, chaque instance est susceptible d'être associée à un ensemble de labels. Une première approche pour répondre à ce problème serait sans doute de diviser le problème multilabels en de multiples problèmes à une classe. Cela aboutirait alors à un classifieur Ignore ou X-Ignore par label, et chaque label serait ainsi prédit séparément. Une seconde approche possible serait de construire des classes "virtuelles", chacunes d'entre elles représentant chaque combinaison de labels. On reviendrait alors de cette manière à un problème de classification à une seule classe. Cette méthode reste cependant très contraignante, car nous pouvons nous retrouver avec une très grande quantité de classes. De plus, dans un contexte multilabels, il est souvent très rare d'avoir beaucoup d'instances ayant exactement les mêmes labels. Il nous faudrait alors un jeu de données avec un très grand nombre d'instances pour que le classifieur généré soit pertinent. Une autre stratégie développée par [Schapire and Singer, 2000] serait de construire une fonction classant les labels les plus probables en haut du classement. Ainsi, de nombreuses méthodes multilabels ont été développées, et il serait intéressant de voir s'il serait possible de les incorporer dans

les modèles Ignore, X-Ignore et ExpertS afin qu'ils puissent répondre au problème de multiples annotateurs en présence de multiples labels.

Tout au long de ce travail nous nous sommes aussi plaçés dans un contexte de problème d'apprentissage hors-ligne. L'apprentissage hors-ligne correspond à la génération d'un classifieur à l'aide d'un jeu de données disponible lors de l'étape d'apprentissage. Ignore, X-Ignore et ExpertS ont bien été générés dans un tel cadre, avec un jeu de validation pour les tester. Or ce type d'apprentissage montre certaines limites. Tout d'abord, les algorithmes issus de l'apprentissage hors-ligne sont généralement réalisables sur des volumes de données de taille moyenne. Au delà, leur temps d'exécution ainsi que la lecture des données peuvent très vite devenir relativement longs, et il devient alors difficile de générer le classifieur. Un autre inconvénient de ce genre d'algorithme est dans le cas où les données ne sont pas toutes immédiatement disponibles. Il arrive très souvent en effet que les données ne soient pas entièrement collectées à un instant donné, mais arrivent en continue. Dans le cas d'apprentissage hors-ligne, l'ajout de données nécessite la regénération entière du classifieur. Un autre type d'algorithme a alors été développé pour répondre à ce problème : l'apprentissage incrémental. Ce dernier permet en effet de recevoir et d'intégrer de nouvelles instances sans avoir à réaliser de nouveau un apprentissage complet. Pour cela, pour un ensemble d'instance $\{x_1, x_2, ..., x_n\}$, l'apprentissage incrémental produit des modèles $\{f_1, f_2, ..., f_n\}$, où chaque modèle f_{i+1} ne dépend que du modèle précédent f_i et de l'instance x_i. Les algorithmes ne lisent donc qu'une seule fois les exemples, et le temps d'apprentissage devient ainsi beaucoup plus rapide que dans un contexte d'apprentissage hors-ligne. De nombreux algorithmes d'apprentissage hors-ligne ont été adaptés au d'apprentissage incrémental : les Séparateurs à Vastes Marges, les réseaux de neurones, la méthode du k plus proche voisin. Il serait intéressant de voir si les trois méthodes proposées dans cette thèse sont adaptables au cas incrémental, lorsque les annotations des experts sont reçues au fur et à mesure, et non en une seule fois. Une recherche rapide sur internet montre par ailleurs l'adaptation de l'algorithme EM utilisé tout au long de cette thèse au cas incrémental [Neal and Hinton, 1999], ce qui peut suggérer une possibilité d'amélioration de nos algorithmes à ce contexte.

Annexe A

Croissance de la Vraisemblance lors des Itérations de l"Algorithme EM

Résultat. On montre que H se dégrade naturellement au cours des itérations de l'algorithme EM.

Preuve. On calcule $\left(H(\Theta^{q+1},\Theta^q) - H(\Theta^q,\Theta^q)\right)$.

$$\begin{aligned}
\Big(H(\Theta^{q+1},\Theta^q) - H(\Theta^q,\Theta^q)\Big) &= \sum_{y\in\mathcal{Y}} p(y|x,\Theta^{q+1}) log(p(y|x,\Theta^q)) - \sum_{y\in\mathcal{Y}} p(y|x,\Theta^q) log(p(y|x,\Theta^q)) \\
&= \sum_{y\in\mathcal{Y}} log\left(\frac{p(y|x,\Theta^{q+1})}{log(p(y|x,\Theta^q))}\right) p(y|x,\Theta^q) \\
&\leq log\ \left(\sum_{y\in\mathcal{Y}} \left(\frac{p(y|x,\Theta^{q+1})}{log(p(y|x,\Theta^q))}\right) p(y|x,\Theta^q)\right) \\
&= log\ \left(\sum_{y\in\mathcal{Y}} p(y|x,\Theta^{q+1})\right) \\
&= log\ 1 \\
&= 0
\end{aligned}$$

Ainsi, maximiser la fonction Q suffit à garantir la croissance de la vraisemblance.

Annexe B

Modèles de mélange : Calcul des Paramètres de Proportion lors de l'Etape M de l'Algorithme EM

Résultat. Afin d'avoir une nouvelle estimation des paramètres de proportion, il nous faut maximiser la fonction Q par rapport à ces paramètres. Or dans le cas de modèles de mélange, Q est donnée par :

$$Q(\Theta, \Theta^q) = \sum_{i=1}^{N} \sum_{k=1}^{K} \mu_{ik}^q log(\pi_k f(x_i; \Theta_k)) \tag{B.1}$$

où

$$\mu_{ik}^q = p(y_i = k|x_i, \Theta^q) = \frac{\pi_k^q f(x_i; \Theta_k^q)}{\sum_{k'=1}^{K} \pi_{k'}^q f(x_i; \Theta_{k'}^q)} \tag{B.2}$$

La maximisation de cette expression donne alors lieu à la solution suivante :

$$\pi_k^{q+1} = \sum_{i=1}^{N} \frac{\mu_{ik}^q}{N} \tag{B.3}$$

Preuve. On a :

$$Q(\Theta, \Theta^q) = \sum_{i=1}^{N} \sum_{k=1}^{K} \mu_{ik}^q \left[log(\pi_k) + log(f(x_i; \Theta_k))\right] \tag{B.4}$$

Or le second terme $log(f(x_i; \Theta_k))$ ne dépend pas de π. Il peut donc être considéré comme une constante. En prenant en considération la contrainte $\sum_{k=1}^{K} \pi_k = 1$, on forme le lagrangien qui donne :

$$l(\pi) = \sum_{i=1}^{N} \sum_{k=1}^{K} \mu_{ik}^q log(\pi_k) + \lambda(1 - \sum_{k=1}^{K} \pi_k) \tag{B.5}$$

où λ est le multiplicateur de Lagrange. Il nous faut alors dériver le lagrangien par rapport aux paramètres de proportion π, ce qui donne :

$$\frac{\partial l(\pi)}{\pi_k} = \frac{\sum_{i=1}^{N} \mu_{ik}^q}{\pi_k} - \lambda, \ \forall k \in \{1, ..., K\} \tag{B.6}$$

La maximisation de Q par rapport à π_k revient alors à trouver les valeurs de π_k pour lesquelles cette expression s'annule, et ce, pour tout k $\in \{1, ..., K\}$. Cela revient ainsi à dire que l'on veut que :

$$\begin{array}{rcl} \frac{\sum_{i=1}^{N} \mu_{i1}^q}{\pi_1} & = & \lambda \\ \vdots & = & \vdots \\ \frac{\sum_{i=1}^{N} \mu_{iK}^q}{\pi_K} & = & \lambda \end{array}$$

On multiplie alors chaque terme par les proportions correspondantes, et on les somme. On obtient alors :

$$\sum_{i=1}^{N} \sum_{k=1}^{K} \mu_{ik}^q = \lambda(\pi_1 + \pi_2 + ... + \pi_K) \tag{B.7}$$

Or on rappelle que $\sum_{k=1}^{K} \pi_k = 1$ et $\sum_{k=1}^{K} \mu_{ik}^q = 1$. D'où finalement $\lambda = N$, ce qui permet d'obtenir la solution B.3.

Annexe C

Régression Linéaire

On effectue ici un bref retour sur la régression linéaire. Le but de la méthode est d'établir un lien entre une variable dépendante Y et une variable indépendante X, pour pouvoir, par la suite, faire des prévisions sur Y lorsque X est mesurée. Il s'agit alors d'expliquer Y par une fonction affine de X. Par exemple, on peut souhaiter prévoir la température en fonction de la pression atmosphérique. Un modèle de régression simple est de la forme

$$Y = \beta_0 + \beta_1 X + \epsilon$$

où Y est la variable aléatoire dépendante,
β_0 et β_1 sont de coefficients (ordonnée à l'origine et pente),
X est la variable indépendante (variable explicative),
ϵ est une erreur aléatoire.

L'estimation des paramètres β_0, β_1 et ϵ est obtenue en maximisant la vraisemblance, sous l'hypothèses que les erreurs sont gaussiennes. Cela revient aussi à minimiser la somme des carrés des écarts entre observations et modèle (moindres carrés). Pour une séquence d'observations $\{x_i, y_i\}_{i=1}^{N}$, le critère des moindres carrés s'écrit :

$$min_{\beta_0,\beta_1} \sum_{i=1}^{N} (y_i - \beta_0 - \beta_1 x_i)^2$$

Or ce système est résolu en mettant la dérivée à zéro par rapport à chacun des paramètres. Une fois les deux paramètres calculés, une estimation de Y peut être calculée de la façon suivante :

$$\hat{y_i} = \beta_0 + \beta_1 x_i$$

Les résidus sont estimés alors comme suit :

$$\epsilon_i = y_i - \hat{y_i}$$

Le lecteur peut se référer à l'ouvrage [Cornillon and Matzner-Lober, 2006] pour davantage d'informations sur cette méthode.

Annexe D

Régression logistique

Appendix D.1 : La transformation logistique

Dans la littérature statistique, la régression logistique a été la première méthode utilisée dans le but de modéliser une variable qualitative binaire. Elle est utilisée dans de nombreux domaines, mais surtout dans le cadre de données médicales, par exemple pour connaitre le statut d'un patient, s'il est infecté ou pas à l'égard de certaines maladies. Ainsi, l'objectif de la régression logistique est de générer un modèle de classification qui prédit ou explique une variable catégorielle à partir d'un ensemble de variables explicatives.
Soit Y_i le statut d'infection pour le i-ème patient. $Y_i = 0$ si le patient est malade, ou $Y_i = 1$ sinon. On note X_i le vecteur composé des p variables explicatives de Y_i. Ainsi, $X_i = \{x_{i1}, ..., x_{ip}\}$. Le modèle de régression logistique modélise alors la relation qui existe entre la variable catégorielle Y_i et le vecteur X_i comme suit :

$$ln\left(\frac{P(Y_i = 1|X_i)}{1 - P(Y_i = 1|X_i)}\right) = \beta^T X_i \tag{D.1}$$

Cette relation qui a été adoptée pour plusieurs raisons. Le rapport $\frac{P(Y_i=1|X_i)}{1-P(Y_i=1|X_i)}$, qui exprime une "côte", est appelé odds et traduit les chances relatives que le patient soit sain. La régression logistique peut être interprétée comme la recherche d'une combinaison linéaire du "'log odds", tandis que les coefficients du modèle expriment des "odds ratio", c'est-à-dire l'influence d'une variable explicative sur la variable à prédire.

Appendix D.2 : Sensibilité, spécificité, courbe ROC

1. Sensibilité α : C'est la proportion de cas, parmi les individus du groupe 1, qui sont bien classés par la procédure. Autrement dit, si X est la vraie classe du sujet et Y la classe prédite par la méthode de classification, on a :

$$\alpha = P(\hat{Y} = 1|X = 1)$$

2. Spécificité β : C'est la proportion de cas, parmi les individus du groupe 0, qui sont bien classés par la procédure. Autrement dit :

$$\beta = P(\hat{Y} = 0|X = 0)$$

3. Courbe ROC : C'est la courbe représentative de la sensibilité en fonction de (1-spécificité).

La courbe ROC est un outil de comparaison de modèles. L'idée est de faire varier le seuil de 0 à 1 et pour chaque cas, de représenter sur un graphique le taux de vrais positifs (sensibilité) en ordonnée et le taux de faux positifs (1-spécificité) (en abscisse). L'aire sous la courbe, appelé AUC (Area Under the roc Curve) est ainsi une mesure du pouvoir prédictif de la variable X.

Appendix D.3 : Extension au cas Multiclasses

Dans le but d'étendre la régression logistique au cas multiclasses, il suffit d'effectuer, pour les K classes disponibles, (K-1) modèles de régression logistique indépendants :

$$ln\left(\frac{P(Y_i = 1|X_i)}{P(Y_i = K|X_i)}\right) = \beta^1 X_i$$

$$ln\left(\frac{P(Y_i = 2|X_i)}{P(Y_i = K|X_i)}\right) = \beta^2 X_i$$

$$...$$

$$ln\left(\frac{P(Y_i = (K-1)|X_i)}{P(Y_i = K|X_i)}\right) = \beta^{K-1} X_i$$

En passant à l'exponentielle, on obtient :

$$P(Y_i = 1|X_i) = P(Y_i = K|X_i)e^{\beta^1 X_i}$$

$$P(Y_i = 2|X_i) = P(Y_i = K|X_i)e^{\beta^2 X_i}$$

$$...$$

$$P(Y_i = (K-1)|X_i) = P(Y_i = K|X_i)e^{\beta^{K-1} X_i}$$

Or la somme des probabilités sur toutes les classes doit être égale à 1, ce qui donne :

$$P(Y_i = K|X_i) = \frac{1}{1 + \sum_{k=1}^{K-1} e^{\beta_k X_i}}$$

Ce qui nous permet finalement d'avoir toutes les autres probabilités :

$$P(Y_i = 1|X_i) = \frac{e^{\beta^1 X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k X_i}}$$

$$P(Y_i = 2|X_i) = \frac{e^{\beta^2 X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k X_i}}$$
$$\dots$$
$$P(Y_i = (K-1)|X_i) = \frac{e^{\beta^{K-1} X_i}}{1 + \sum_{k=1}^{K-1} e^{\beta_k X_i}}$$

Pour de plus amples informations sur la régression logistique, le lecteur peut se référer aux ouvrages suivants [Hastie et al., 2001, Saporta, 2011].

Annexe E

Résultats Ignore

Appendix E.1 : Ignore Binaire et Incertitude Totale

Nous présentons ci-dessous les graphiques obtenus pour les autres jeux de données synthétiques de l'UCI Machine Learning Repository, pour la première contribution Ignore.

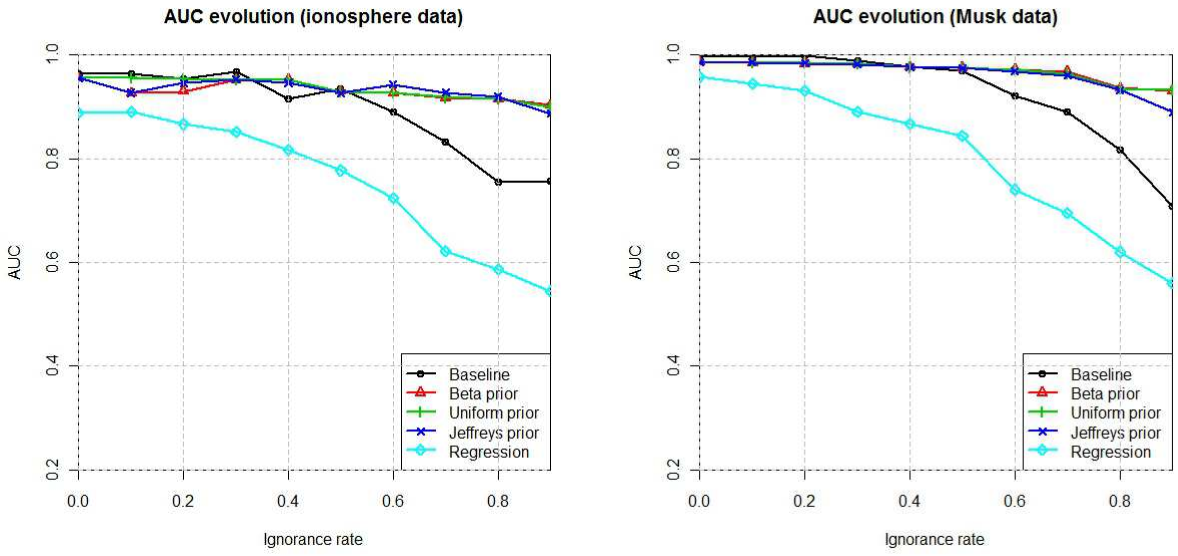


FIGURE E.1 – Ignore Binaire et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression linéaire, en fonction du taux d'incertitude des annotateurs.

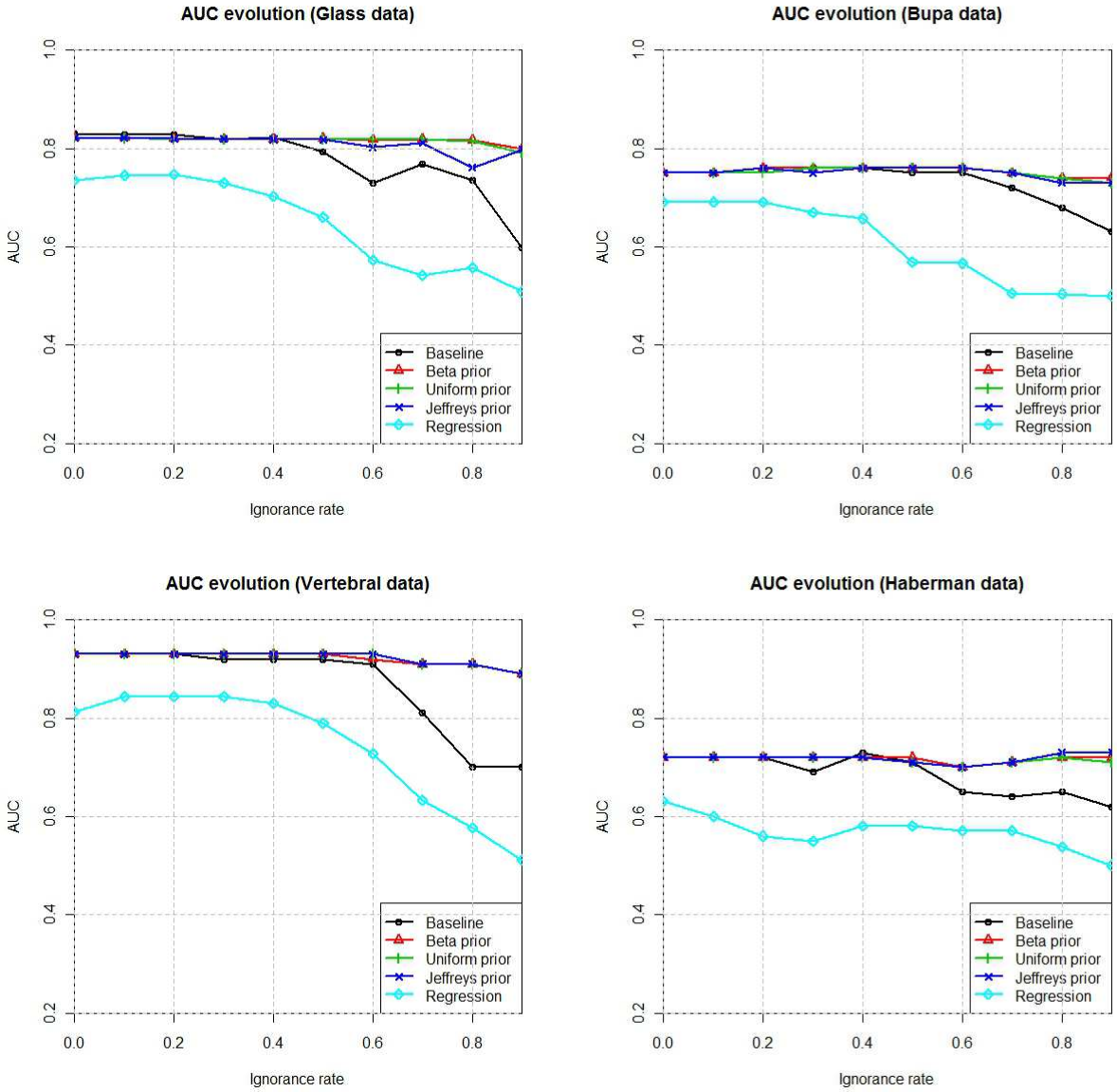


FIGURE E.2 – Ignore Binaire et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression linéaire, en fonction du taux d'incertitude des annotateurs.

Appendix E.2 : Ignore Binaire et Incertitude Partielle

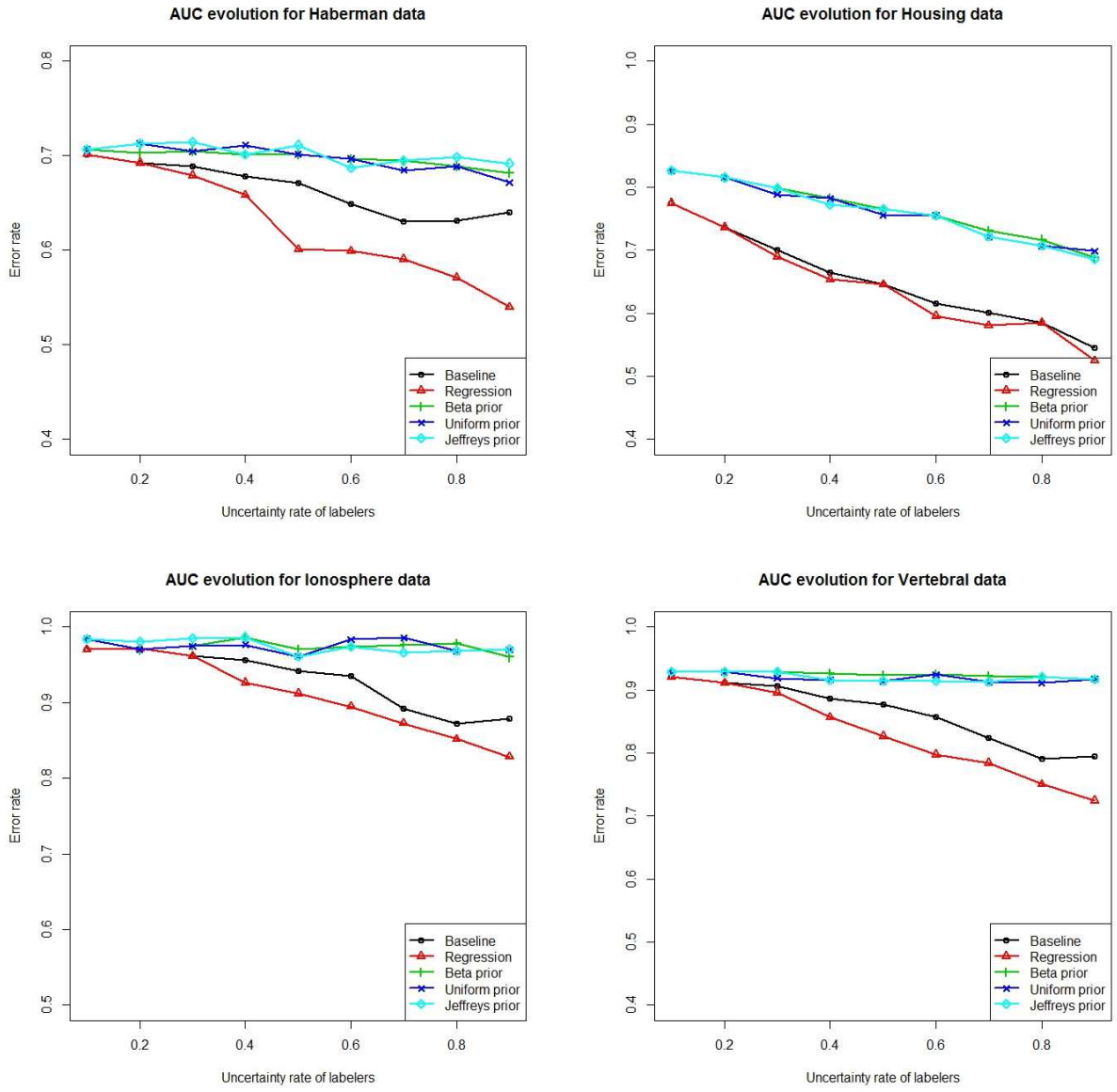


FIGURE E.3 – Ignore Binaire et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression linéaire, en fonction du taux d'incertitude des annotateurs.

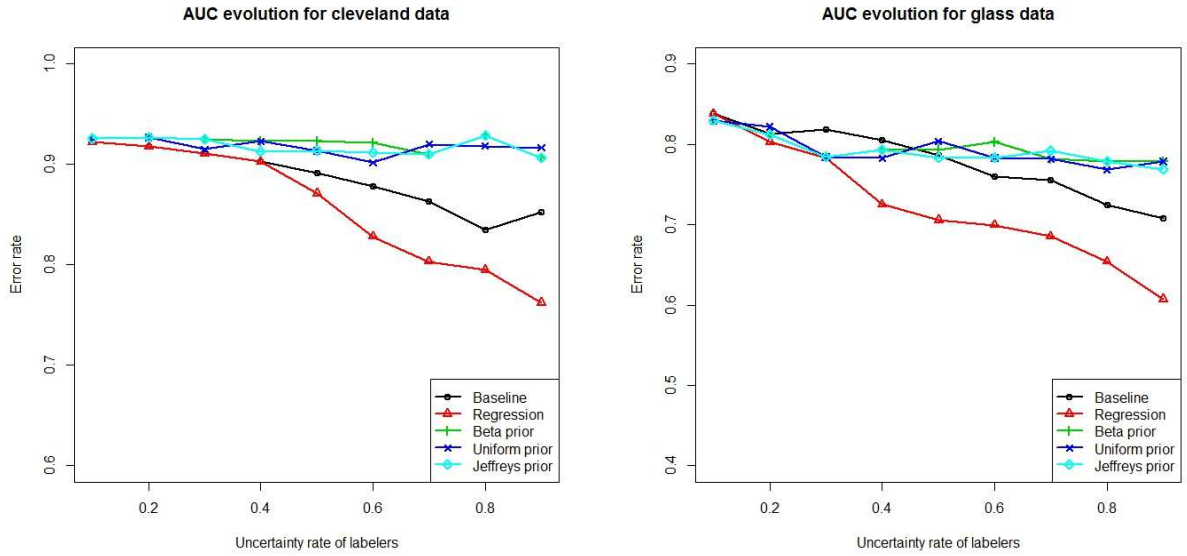


FIGURE E.4 – Ignore Binaire et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression linéaire, en fonction du taux d'incertitude des annotateurs.

Appendix E.3 : Ignore Multiclasses et Incertitude Totale

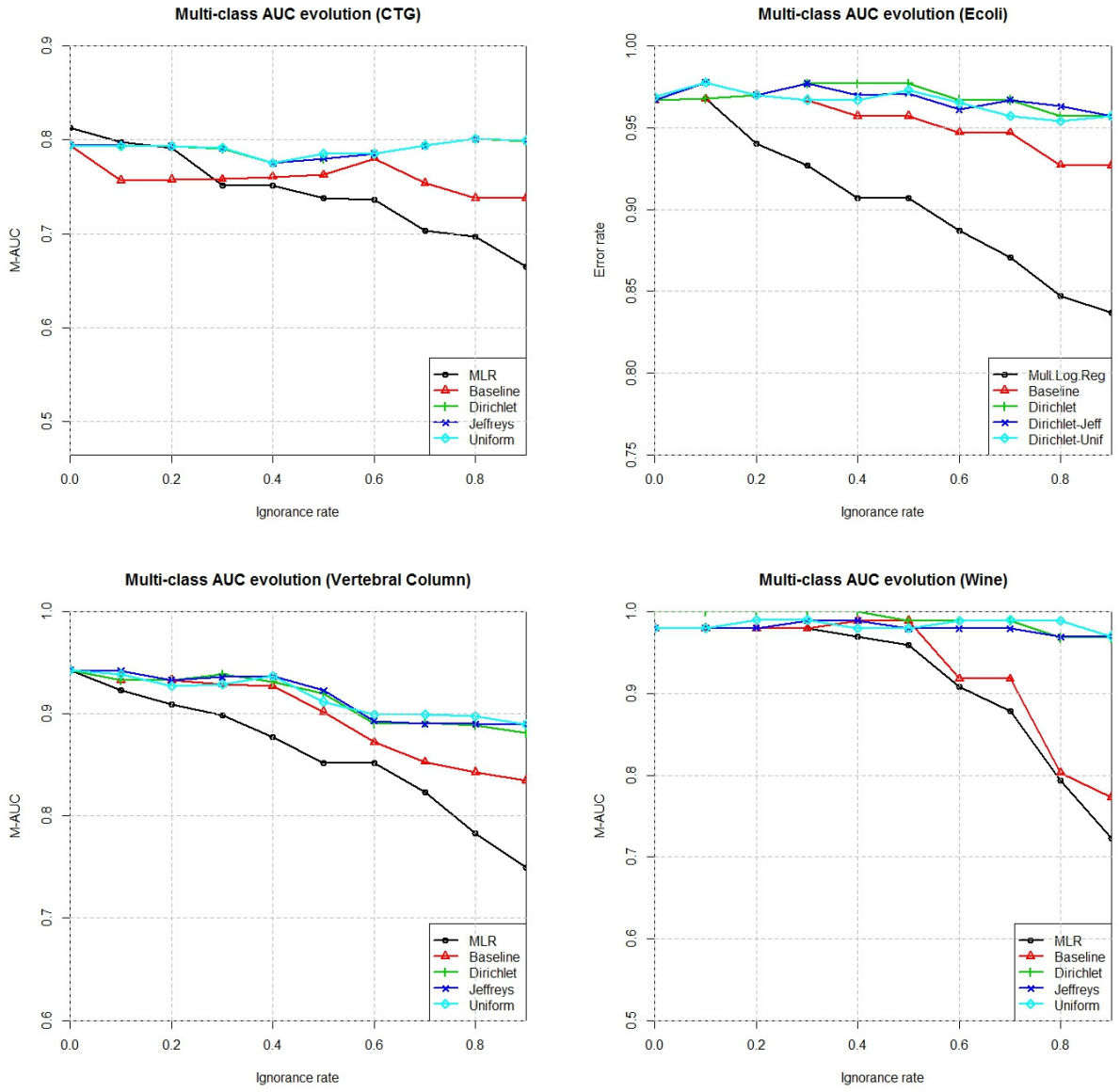


FIGURE E.5 – Cas Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

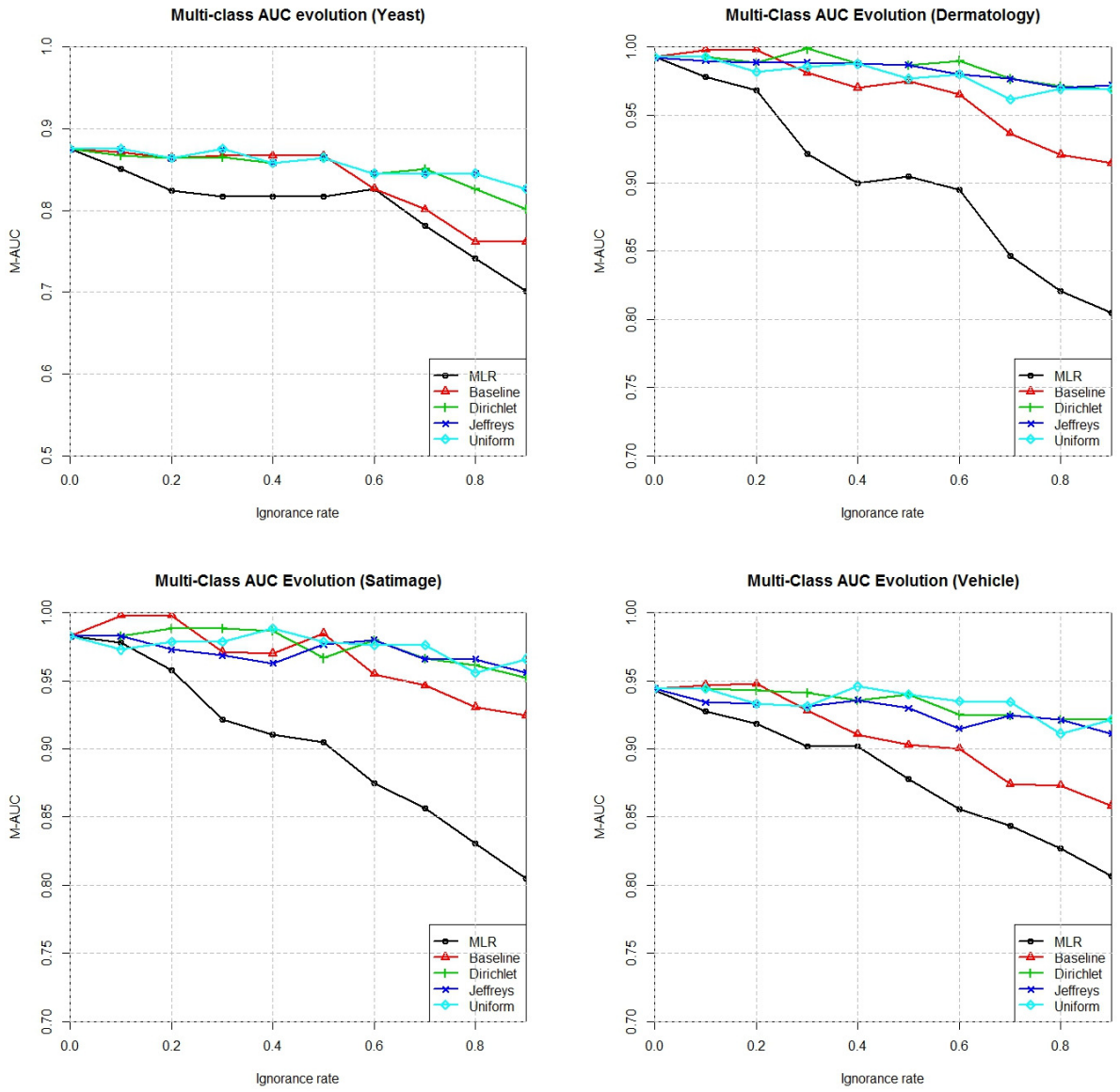


FIGURE E.6 – Cas Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

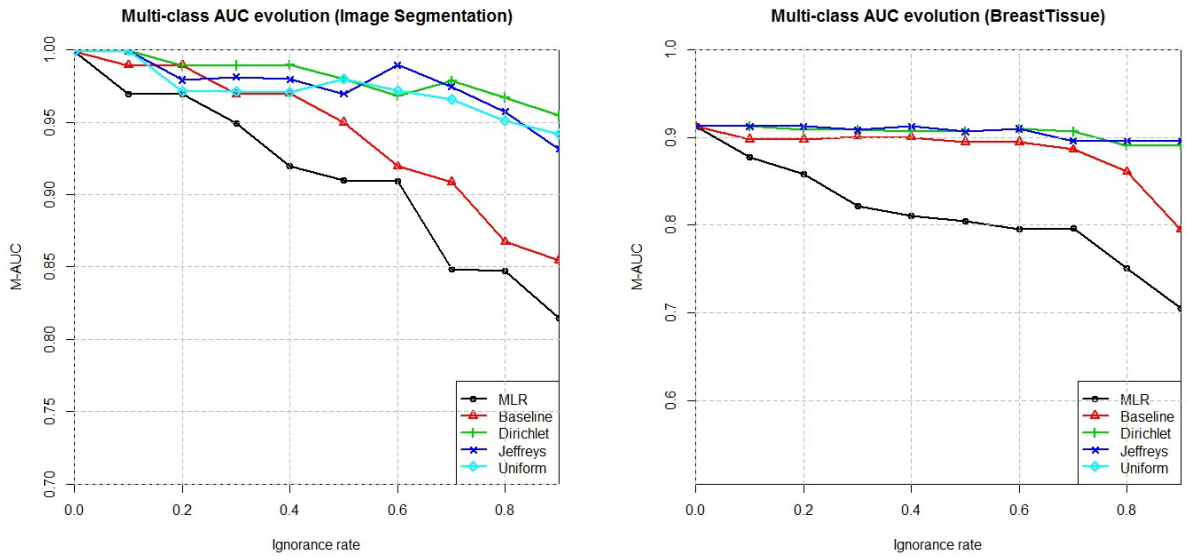


FIGURE E.7 – Cas Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

Appendix E.4 : Ignore Multiclasses et Incertitude Partielle

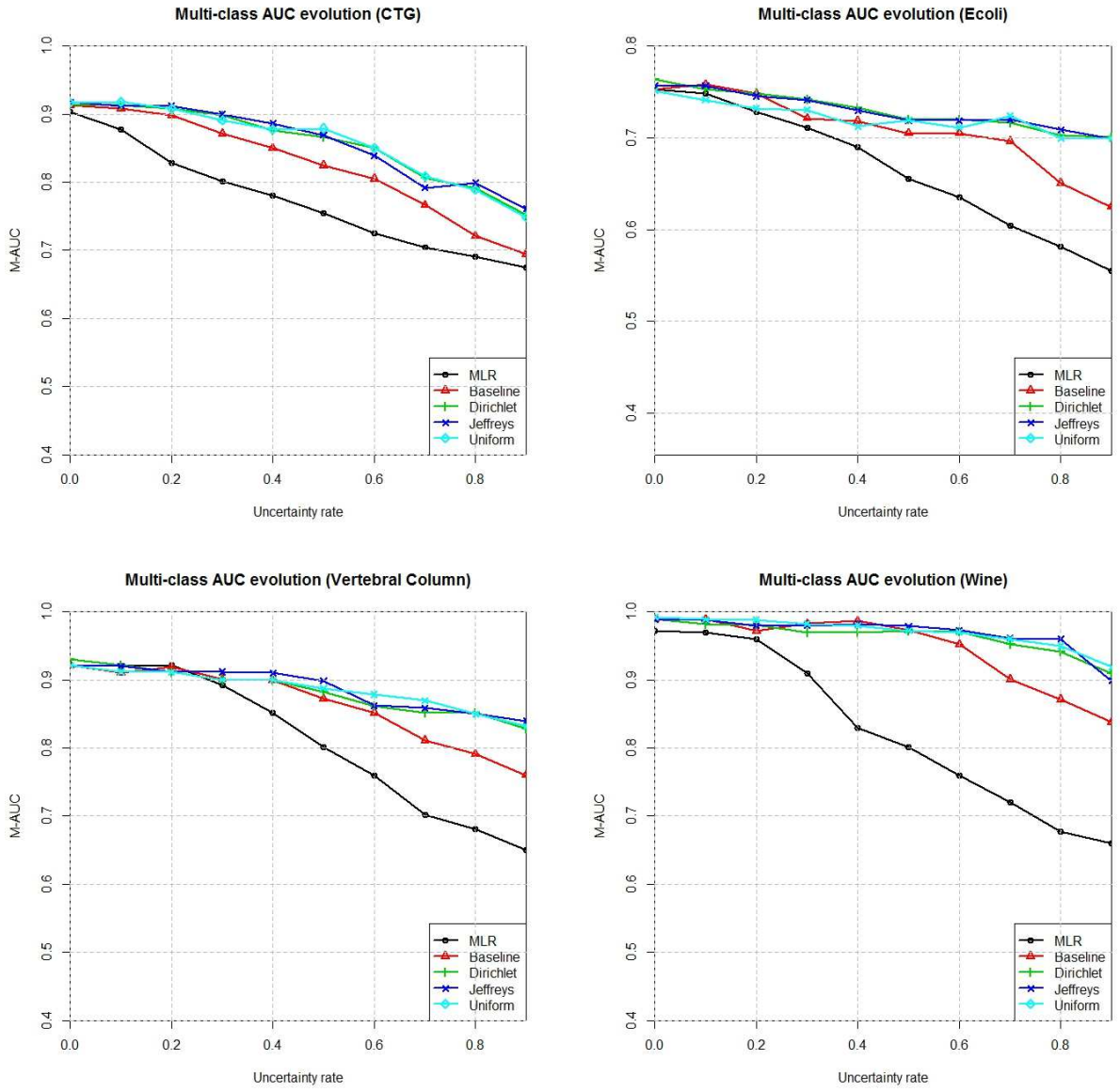


FIGURE E.8 – Cas Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

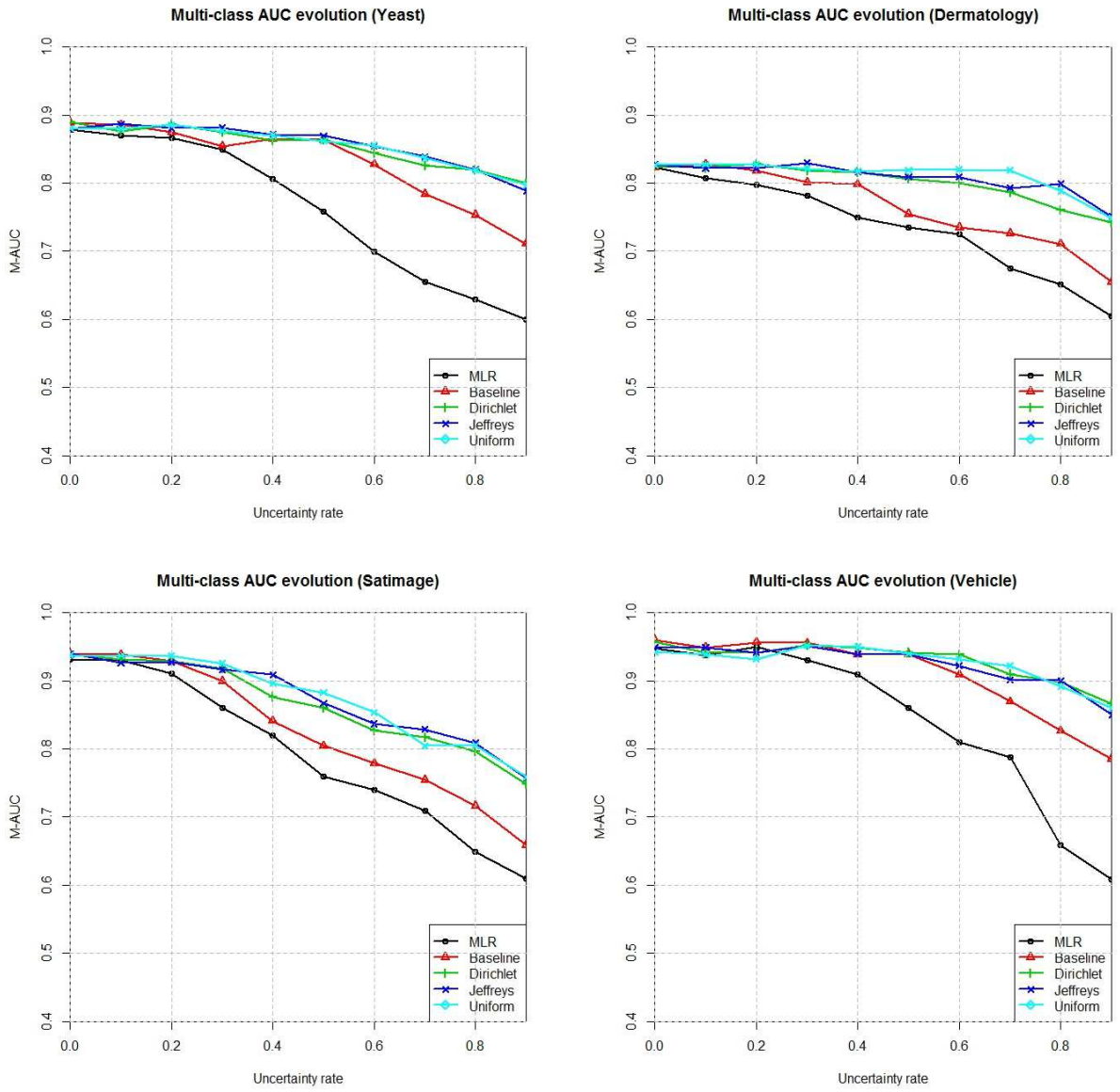


FIGURE E.9 – Cas Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

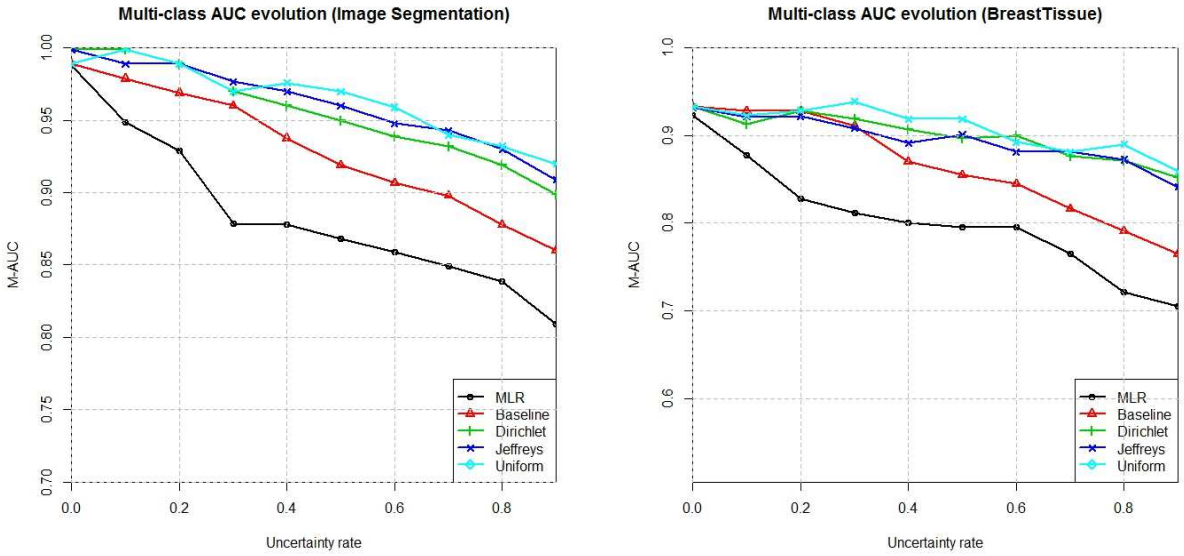


FIGURE E.10 – Cas Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre Ignore, Raykar et la régression logistique multimodale, en fonction du taux d'incertitude des annotateurs.

Annexe F

Résultats X-Ignore

Nous présentons ci-dessous les graphiques obtenus pour les autres jeux de données synthétiques de l'UCI Machine Learning Repository, pour la deuxième contribution X-Ignore.

Appendix F.1 : X-Ignore Binaire et Incertitude Totale

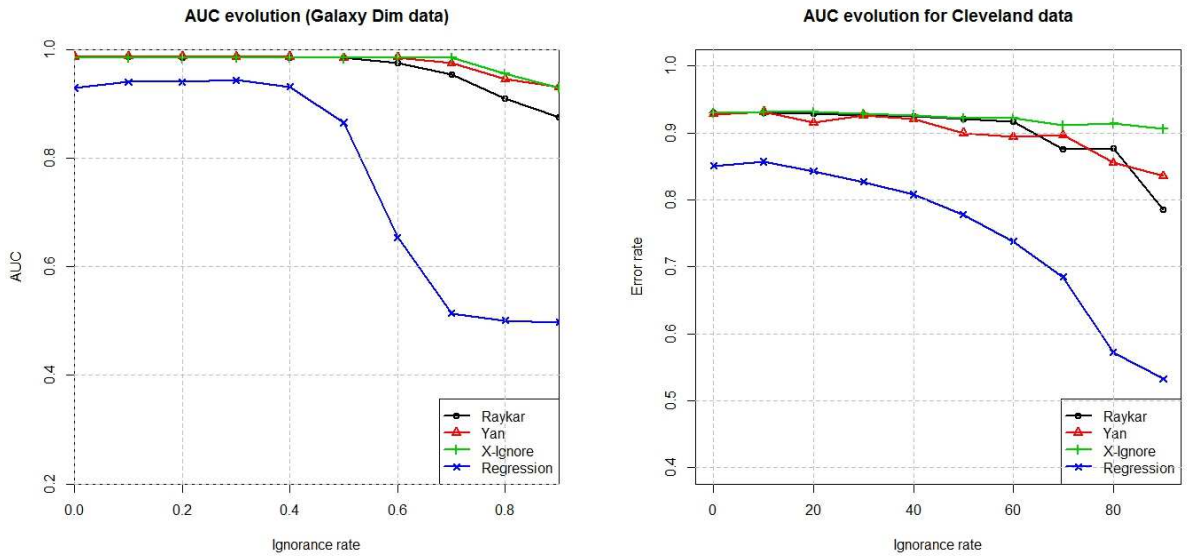


FIGURE F.1 – Cas Binaire et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire en fonction du taux d'incertitude des annotateurs.

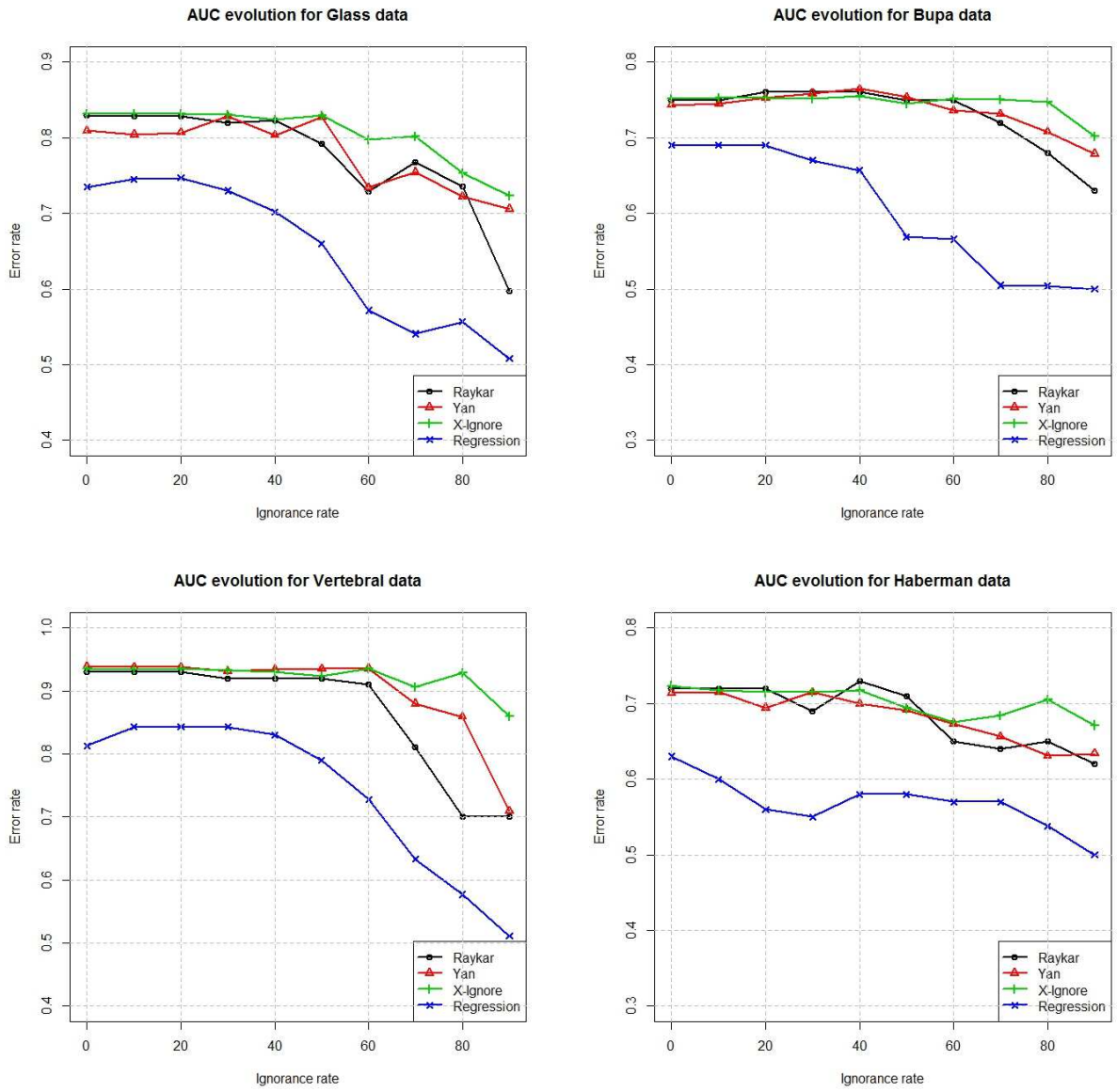


FIGURE F.2 – Cas Binaire et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire en fonction du taux d'incertitude des annotateurs.

Appendix F.2 : X-Ignore Binaire et Incertitude Partielle

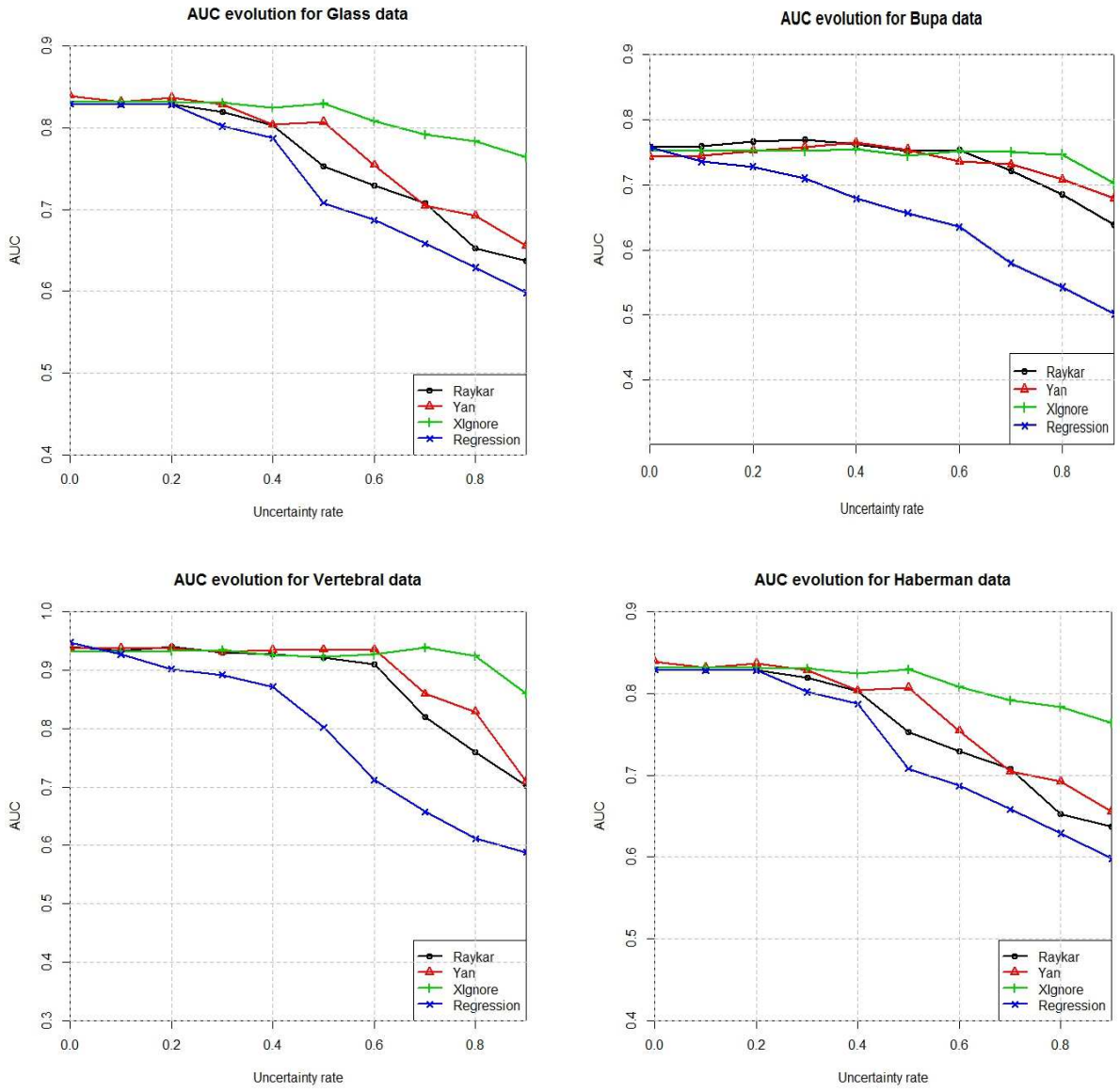


FIGURE F.3 – Cas Binaire et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire en fonction du taux d'incertitude des annotateurs.

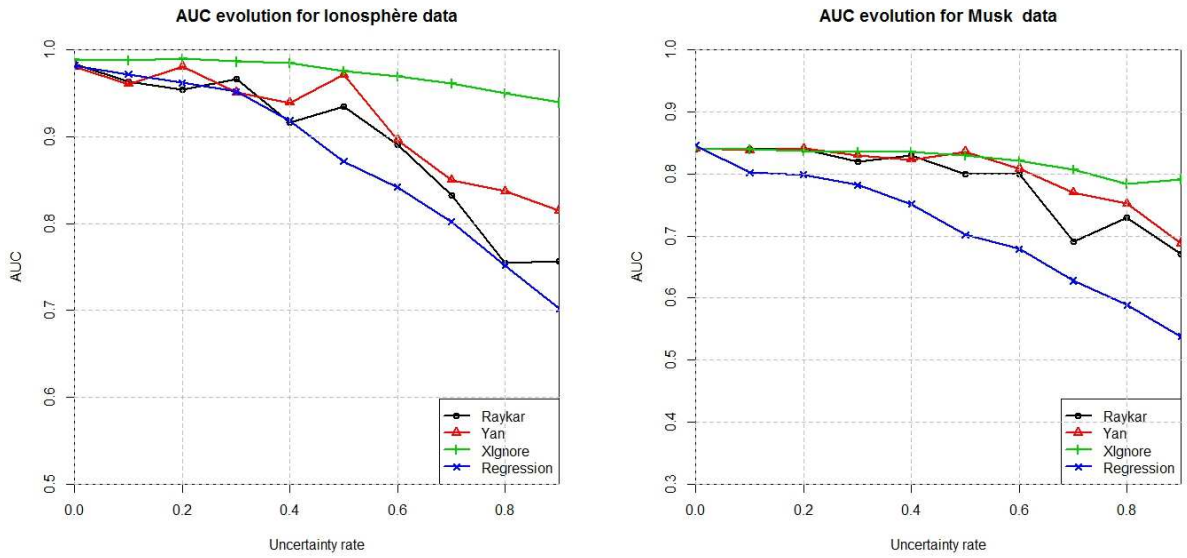


FIGURE F.4 – Cas Binaire et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire en fonction du taux d'incertitude des annotateurs.

Appendix F.3 : X-Ignore Multiclasses et Incertitude Totale

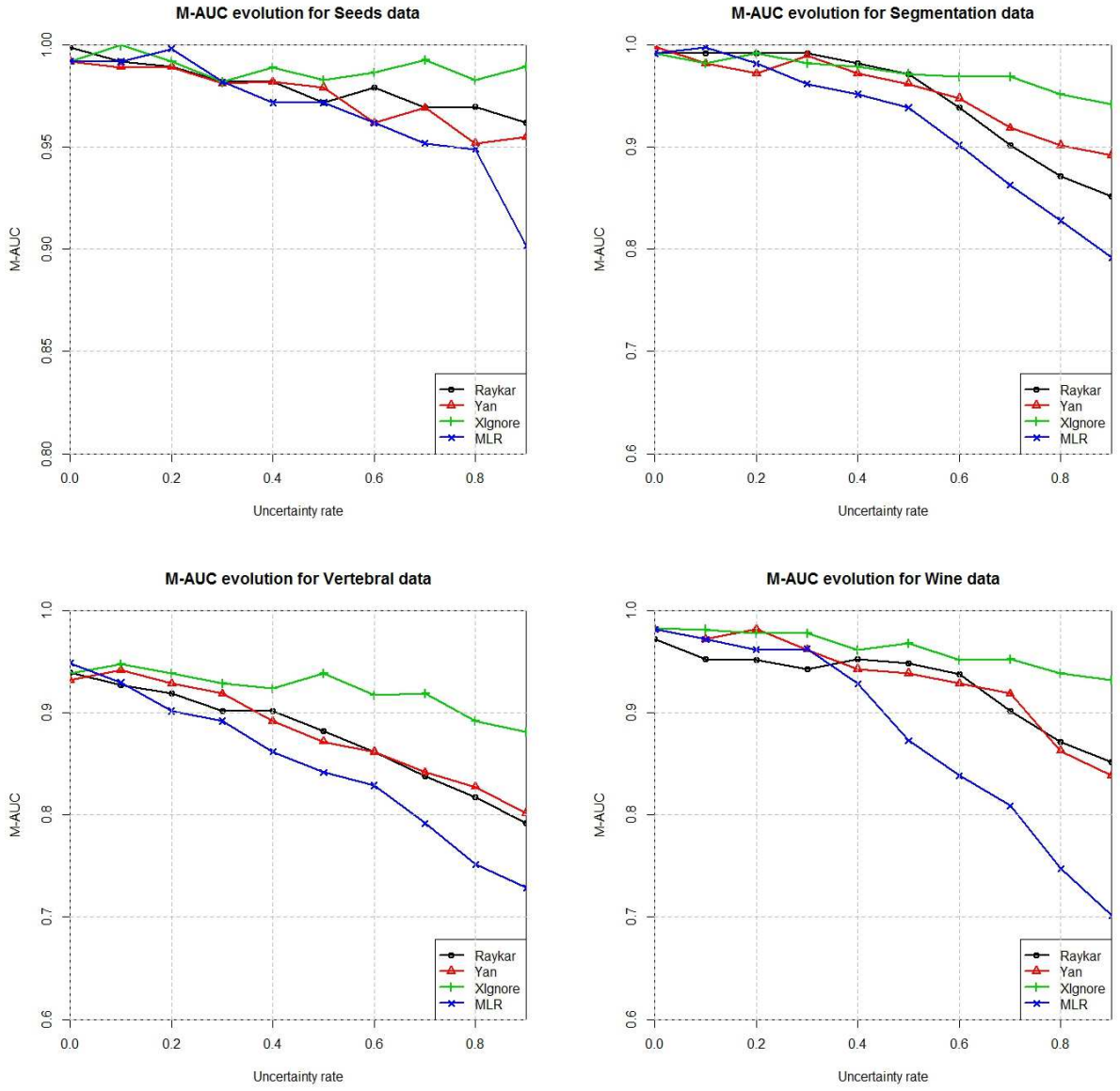


FIGURE F.5 – Cas Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

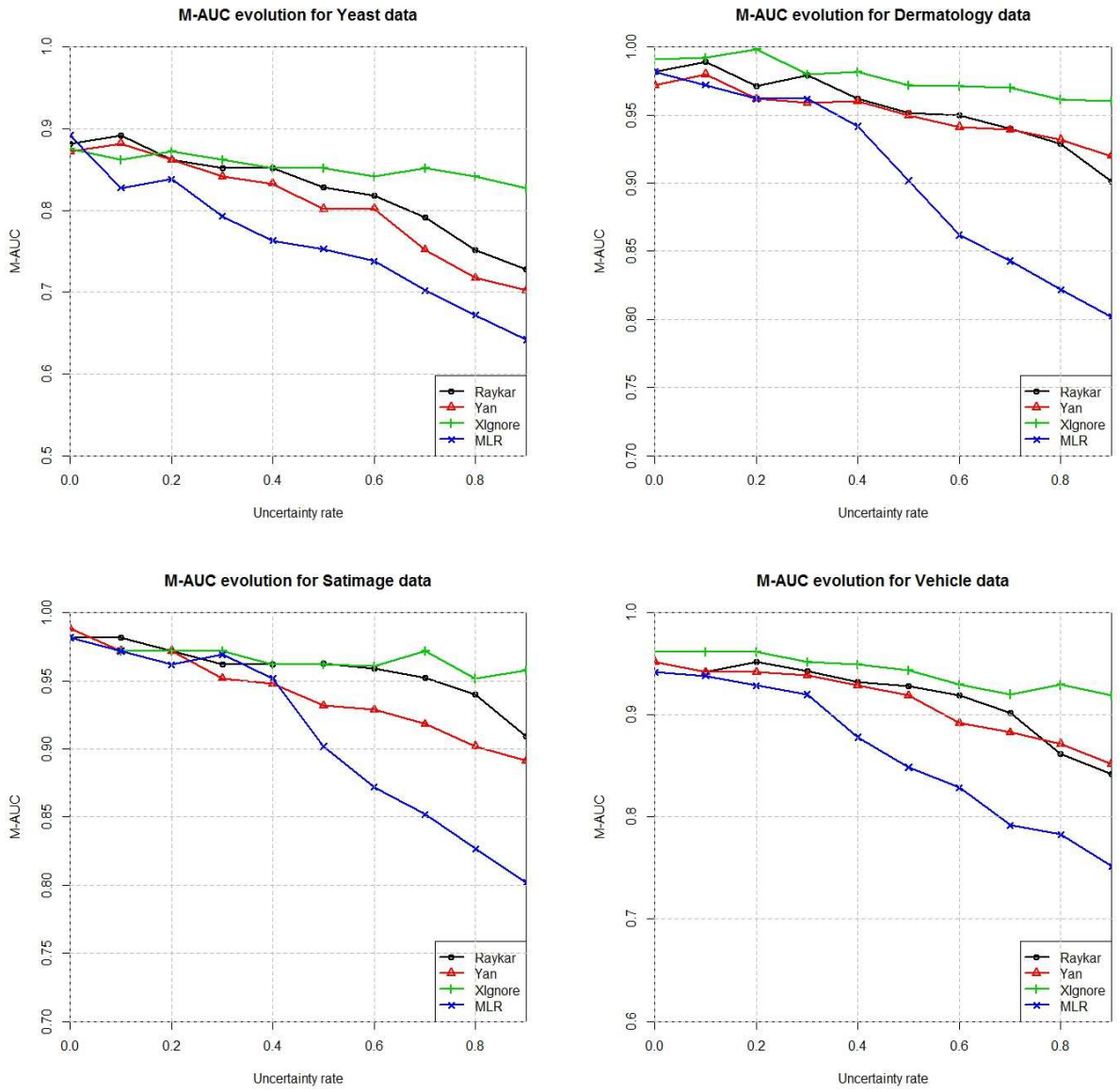


FIGURE F.6 – Cas Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

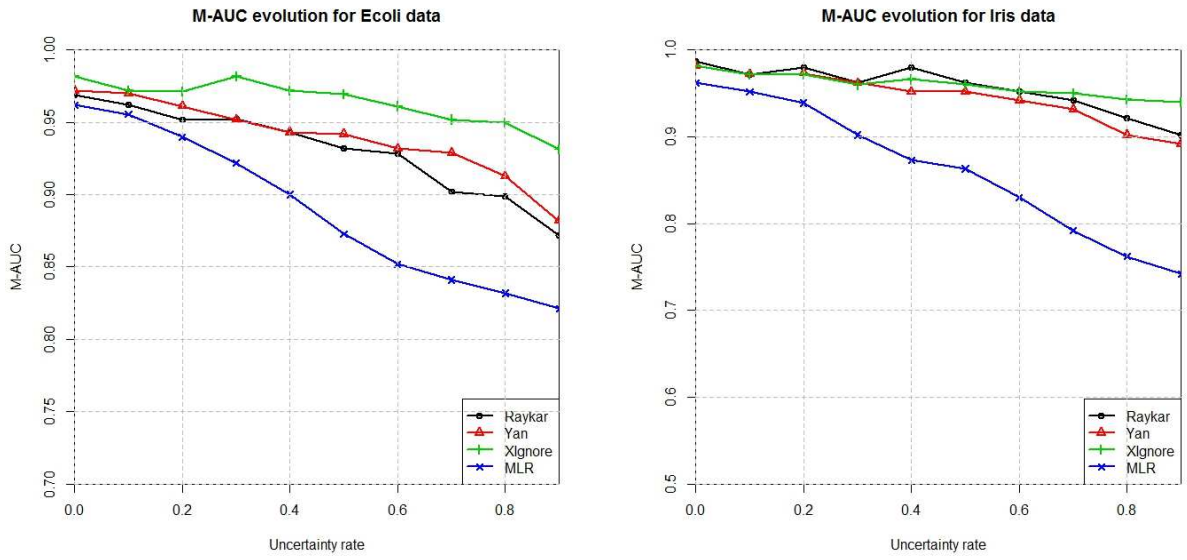


FIGURE F.7 – Cas Multiclasses et Incertitude Totale : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

Appendix F.4 : Ignore Multiclasses et Incertitude Partielle

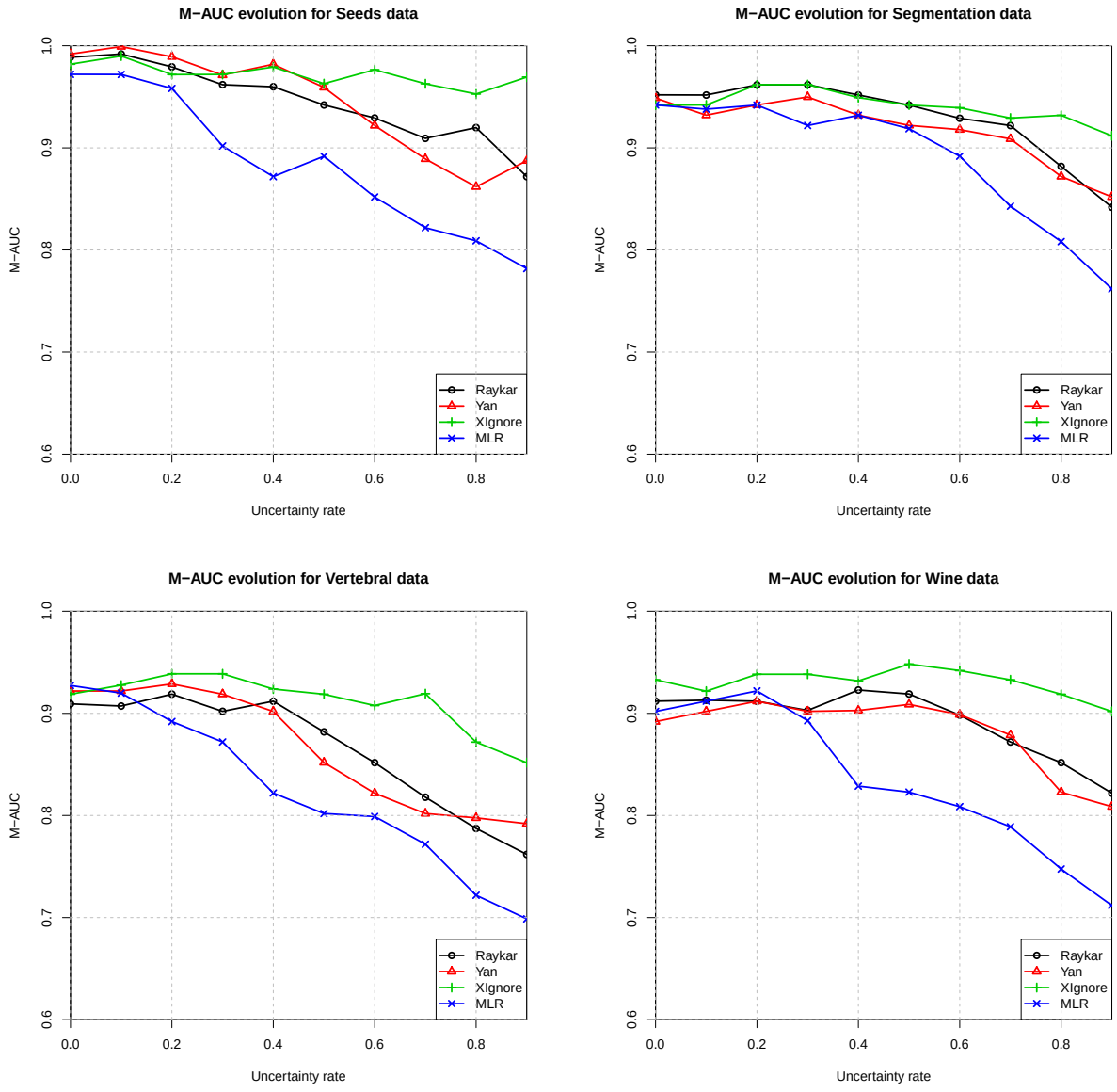


FIGURE F.8 – Cas Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

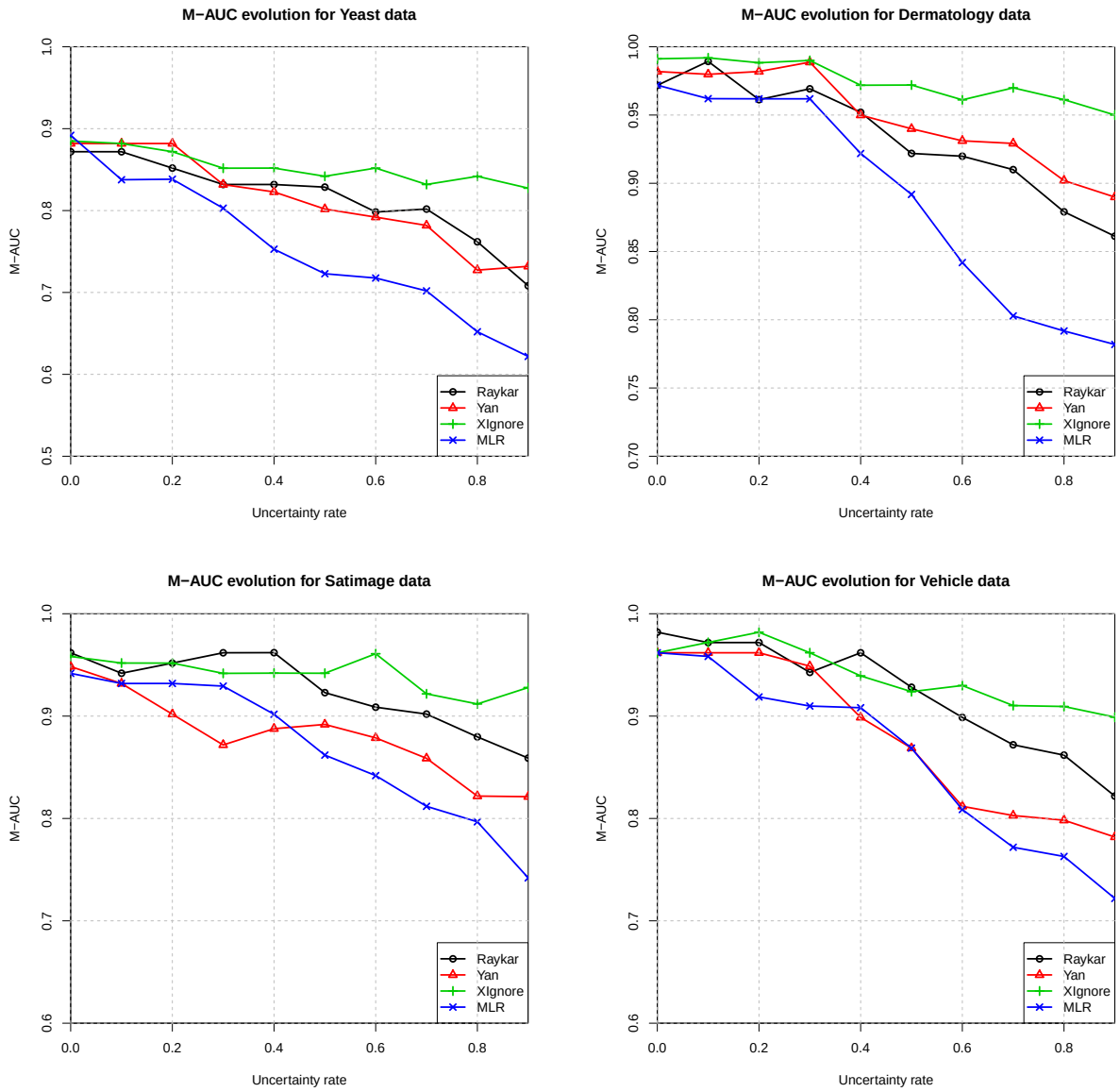


FIGURE F.9 – Cas Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

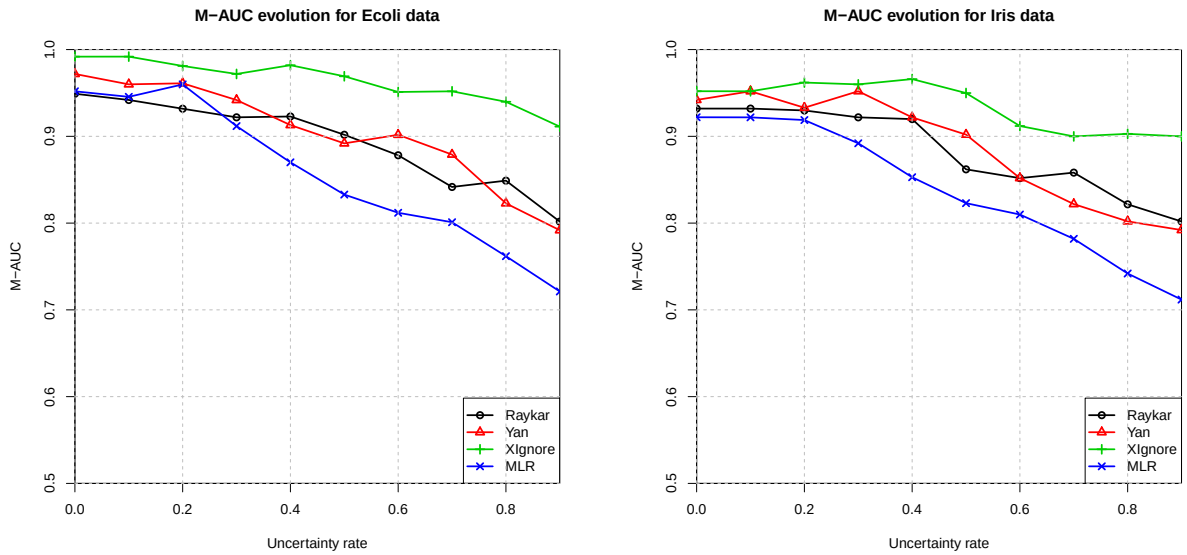


FIGURE F.10 – Cas Multiclasses et Incertitude Partielle : Comparaison de l'évolution de l'AUC entre X-Ignore, Raykar, Yan et la régression linéaire multimodale en fonction du taux d'incertitude des annotateurs.

Annexe G

Rappels sur l'ACP et la S-ACP

L'analyse en composantes principales (ACP) [Pearson, 1901] est une méthode statistique consistant à transformer certaines variables liées entre elles dans un jeu de données par de nouvelles variables décorrélées nommées "composantes principales". Cette transformation a principalement pour objectif d'étudier les liaisons entre les variables, et de simplifier le jeu de données puisqu'une réduction de variables aura lieu, permettant de rendre l'information moins redondante. Nous résumons ici les principaux points théoriques de cette méthode, le lecteur peut retrouver des exposés plus détaillés dans les ouvrages de [Lebart et al., 2000, Jolliffe, 2002].

Appendix G.1 : Quelques Eléments Mathématiques de l'ACP

Soit un jeu de données représenté sous la forme d'un tableau ou d'une matrice X à n lignes et p colonnes :

$$X = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ x_{n1} & \cdots & \cdots & x_{np} \end{pmatrix}$$

avec $x_{i,j}$ la valeur pour la i-ème instance de la variable j.

Le but est de réduire les p variables explicatives du jeu de données X initial, à q variables explicatives, où $q < p$. Ces q variables seront alors appelées les composantes principales, et une nouvelle matrice Z sera alors créee de telle sorte que :

$$Z = \begin{pmatrix} z_{11} & z_{12} & \cdots & z_{1q} \\ z_{21} & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ z_{n1} & \cdots & \cdots & z_{nq} \end{pmatrix}$$

Z est appelée matrice de scores, et ses composantes z^j (les composantes principales) doivent répondre à certaines propriétés :

- Elles doivent être indépendantes entre elles : $cor(Z^k, Z^m) = 0 \ \forall k \ \neq \ m$,
- Chaque composante Z^k doit être une combinaison linéaire des variables X^k :

$$Z^k = \sum_{j=1}^{P} v_j^k X^j$$

- Les élements de cette nouvelle matrice Z doivent restituer le maximum d'information contenue initialement dans la matrice X.

Ainsi, on propose la définition de l'ACP suivante :

Definition 9. *On considère p variables centrées $X^1, X^2, ..., X^p$ observées sur n individus. L'ACP de X correspond à la recherche des q combinaisons linéaires normées de X^j, non corrélées et dont l'inertie (ou la somme des variances) est maximale.*

Pour déterminer la première composante principale Z^1 de X, le problème revient alors à trouver une droite D engendrée par un vecteur unitaire $\vec{V_1}$, telle que cette droite restitue le maximum de l'information initiale. Or les projections des observations sur la droite D sont données par le produit scalaire $d = X\vec{V_1}$, et leur inertie correspond à la somme des carrés de ces projections, en d'autres termes $\vec{V_1}^T(X^TX)\vec{V_1}$.
$\vec{V_1}$ est donc choisi en maximisant l'expression $\vec{V_1}^{'} X^{'} X\vec{V_1}$, sous la contrainte $\vec{V_1}\vec{V_1}^T = 1$. Or ceci revient à un problème d'optimisation classique qui peut être résolu par la méthode de Lagrange, se référer au livre de [Bertsekas, 1996] pour davantage d'informations concernant cette méthode.

Une fois toutes les composantes ACP identifiées, nous devons choisir le nombre de composantes que l'on souhaite inclure dans notre modèle de sorte que l'on obtienne le meilleur modèle possible tout en limitant le nombre de variables au maximum. De nombreux critères de choix ont été proposés dans la littérature (règle de Kaiser, part d'inertie, ect). Le lecteur se réferera aux livres de [Lebart et al., 2000, Jolliffe, 2002] pour de plus amples informations.

On remarque que dans l'ACP, chaque composante principale représente une combinaison linéaire de l'ensemble des variables de départ. Dans le cas où le nombre de variables de départ est très grand, l'interprétation des résultats risque d'être difficile et une alternative, l'ACP Sparse, a alors été proposée.

Appendix G.2 : ACP Sparse

Le but de l'ACP sparse est d'obtenir des composantes facilement interprétables même dans un contexte de données de grande dimension. Pour atteindre cet objectif, la sparse

ACP impose des contraintes supplémentaires qui vont sacrifier de la variance afin de privilégier l'interprétabilité. Il existe plusieurs versions d'ACP sparse, les plus importantes étant [Jolliffe et al., 2003, Zou et al., 2004, Shen and Huang, 2008].
Alors que l'ACP cherche les combinaisons linéaires des variables initiales de X, de sorte que la variance $\vec{V_1}^T(X^TX)\vec{V_1}$ soit maximale sous la contrainte $\vec{V_1}\vec{V_1}^T = 1$, l'ACP sparse ajoute une contrainte supplémentaire. Par exemple, dans [Jolliffe et al., 2003], la contrainte supplémentaire correspond à une contrainte de type Lasso, telle que :

$$\sum_{j=1}^{p} |v_j^k| \leq t$$

avec t un paramètre de régularisation, et v_j^k le k-ième élement du vecteur v_j. L'inconvénient de cette méthode est que le résultat obtenu dépend essentiellement du choix de la valeur de t, qui est un choix très délicat. Un autre inconvénient est que le problème n'est pas convexe, ce qui peut poser des difficultés lors de l'obtention du maximum global.
Dans [Shen and Huang, 2008], les auteurs introduisent plutôt une pénalité de type elastic net pour reproduire la sparsité. Cette méthode permet de fixer certains coefficients à zéro, et de réduire ainsi le nombre de variables explicatives.

Bibliographie

[Abe, 2002] Abe, S. (2002). Analyse of Support Vector Machines. *Neural Networks for Signal Processing XII*, pages 89–98.

[Acosta et al., 2013] Acosta, M., Zaveri, A., Simperl, E., Kontokostas, D., Auer, S., and Lehmann, J. (2013). Crowdsourcing linked data quality assessment. In *12th International Semantic Web Conference, 21-25 October 2013, Sydney, Australia.*

[Akkaya et al., 2010] Akkaya, C., Conrad, A., Wiebe, J., and Mihalcea, R. (2010). Amazon mechanical turk for subjectivity word sense disambiguation. In *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, pages 195–203, Stroudsburg, PA, USA.

[Asuncion and Newman, 2007] Asuncion, A. and Newman, D. (2007). UCI machine learning repository.

[Ballerini et al., 2012] Ballerini, L., Fisher, R. B., Aldridge, B., and Rees, J. (2012). Non-melanoma skin lesion classification using colour image data in a hierarchical K-NN classifier. In *ISBI*, pages 358–361. IEEE.

[Baranger, 1977] Baranger, J. (1977). Introduction à l'analyse numérique. Hermann.

[Bardos, 2001] Bardos, M. (2001). Analyse discriminante. *Dunod.*

[Bayes, 1763] Bayes, T. (1763). An essay towards solving a problem in the doctrine of chances. *Philosophical Transactions of the Royal Society of London*, 53 :370–418.

[Bernardo and Smith, 1994] Bernardo, J. M. and Smith, A. F. (1994). *Bayesian Theory.* Wiley.

[Bertsekas, 1996] Bertsekas, D. P. (1996). *Constrained Optimization and* L*agrange Multiplier Methods (Optimization and Neural Computation Series).* Athena Scientific.

[Bickel and Levina, 2004] Bickel, P. J. and Levina, E. (2004). Some theory for Fisher's linear discriminant function, 'naive Bayes', and some alternatives when there are many more variables than observations. *Bernoulli*, 10 :989–1010.

[Biernacki et al., 2003] Biernacki, C., Celeux, G., and Govaert, G. (2003). Choosing starting values for the EM algorithm for getting the highest likelihood in multivariate gaussian mixture models. *Computational Statistics & Data Analysis*, 41(3-4) :561–575.

[Bishop, 2007] Bishop, C. M. (2007). *Pattern Recognition and Machine Learning (Information Science and Statistics).* Springer, 1 edition.

[Blanzieri and Bryl, 2008] Blanzieri, E. and Bryl, A. (2008). A survey of learning-based techniques of email spam filtering. *Artif. Intell. Rev.*, 29 :63–92.

[Bonnans et al., 2006] Bonnans, J. F., Gilbert, Lemaréchal, C., and Sagastizàbal, C. (2006). *Numerical Optimization – Theoretical and Practical Aspects*. Universitext. Springer Verlag, Berlin.

[Bordes et al., 2005] Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *J. Mach. Learn. Res.*, 6 :1579–1619.

[Boser et al., 1992] Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory*, pages 144–152. ACM Press.

[Bouchon-Meunier and Marsala, 2003] Bouchon-Meunier, B. and Marsala, C. (2003). *Logique floue, Principes, Aide à la Décision*. Hermes.

[Bourbonnais and Terraza, 1998] Bourbonnais, R. and Terraza, M. (1998). *Analyse des Séries Temporelles en Economie*. PUF.

[Box and Tiao, 1973] Box, G. E. P. and Tiao, G. C. (1973). Bayesian inference in statistical analysis. *Wiley Classics Library Edition,John Wiley & Sons*.

[Breiman et al., 1984] Breiman, L., Friedman, J., Olshen, R., and Stone, C. (1984). *Classification and Regression Trees*. Wadsworth and Brooks, Monterey, CA.

[Burges, 1998] Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2 :121–167.

[Chapelle, 2007] Chapelle, O. (2007). Training a support vector machine in the primal. *Neural Comput.*, 19(5) :1155–1178.

[Cornillon and Matzner-Lober, 2006] Cornillon, P.-A. and Matzner-Lober, E. (2006). *Régression. Théorie et Applications*. Springer.

[Crammer and Singer, 2003] Crammer, K. and Singer, Y. (2003). A new family of online algorithms for category ranking. *Journal of Machine Learning Research*, 3 :1025–1058.

[Davison and Hinkley, 1997] Davison, A. C. and Hinkley, D. V. (1997). *Boostrap methods and their applications*. Cambridge University Press.

[Dawid and Skene, 1979] Dawid, A. P. and Skene, A. M. (1979). Maximum likelihood estimation of observer error-rates using the EM algorithm. *Applied Statistics*, 28 :20–28.

[Dempster et al., 1977] Dempster, A., Laird, N., and Rubin, D. (1977). Maximum likelihood estimation from incomplete data. *J. of the Royal Statistical Society (B)*, 39(1).

[Donmez and Carbonell, 2008] Donmez, P. and Carbonell, J. G. (2008). Proactive learning : Cost-sensitive active learning with multiple imperfect oracles. In *CIKM*, pages 619–628. ACM.

[Dubois and Prade, 1985] Dubois, D. and Prade, H. (1985). *Fuzzy sets and systems, Theory and applications*. Academic Press.

[Dubois and Prade, 1988] Dubois, D. and Prade, H. (1988). Representation and combination of uncertainty with belief functions and possibility measures. *Computational Intelligence*, 4 :244–264.

[Dudoit et al., 2002] Dudoit, S., Fridlyand, J., and Speed, T. (2002). Comparison of discrimination methods for the classification of tumors using gene expression data. *Journal of the American Statistical Association*, 457 :77–87.

[Efron, 1979] Efron, B. (1979). Bootstrap methods : Another look at the Jackknife. *The Annals of Statistics*, 7(1) :1–26.

[Efron and Tibshirani, 1993] Efron, B. and Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, New York, NY.

[Elkan, 2001] Elkan, C. (2001). *The Foundations of Cost-Sensitive Learning*. IJCAI'01 Proceedings of the 17th International Joint Conference on Artificial Intelligence.

[Fan and Lv, 2006] Fan, J. and Lv, J. (2006). Sure independence screening for ultra-high dimensional feature space. *arXiv*.

[Fan et al., 2008] Fan, R.-E., Chang, K.-W., Hsieh, C.-J., Wang, X.-R., and Lin, C.-J. (2008). LIBLINEAR : A library for large linear classification. *Journal of Machine Learning Research*, 9 :1871–1874.

[Fang et al., 2012] Fang, M., Zhu, X., Li, B., 0003, W. D., and Wu, X. (2012). Self-taught active learning from crowds. In Zaki, M. J., Siebes, A., Yu, J. X., Goethals, B., Webb, G. I., and Wu, X., editors, *ICDM*, pages 858–863. IEEE Computer Society.

[Figueiredo, 2002] Figueiredo, M. (2002). Adaptive sparseness using jeffreys prior. In *Proc. Advances in Neural Information Processing Systems 14*, pages 697–704. MIT Press.

[Figueiredo and Nowak, 2001] Figueiredo, M. A. T. and Nowak, R. (2001). Wavelet-based image estimation : An empirical bayes approach using Jeffreys' noninformative prior. volume 10, pages 1322–1331.

[Fink et al., 2011] Fink, E., Sharifi, M., and Carbonell, J. (2011). Application of machine learning and crowdsourcing to detection of cybersecurity threats. *Proceedings of the DHS Science Confernece, Fifth Annual University Network Summit*.

[Fisher, 1922] Fisher, R. A. (1922). On the mathematical foundations of theoretical statistics. *Philosophical Transactions of the Royal Society of London. Series A, Containing Papers of a Mathematical or Physical Character*, 222 :309–368.

[Fix and Hodges, 1951] Fix, E. and Hodges, J. L. (1951). Discriminatory analysis, nonparametric discrimination : Consistency properties. *US Air Force School of Aviation Medicine*, Technical Report 4 :477+.

[Friedman, 1996] Friedman, J. H. (1996). Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, Stanford, CA.

[Funtowicz and Ravetz, 1990] Funtowicz, S. and Ravetz, J. (1990). Uncertainty and quality in science for policy. *Ecological Economics*.

[Gacôgne, 1997] Gacôgne, L. (1997). *Eléments de Logique Floue*. Hermes.

[Gareau et al., 2012] Gareau, D., Hennessy, R., and Jacques, S. (2012). Automated detection of melanoma. *Google Patents*.

[Gelman, 2006] Gelman, A. (2006). Prior distributions for variance parameters in hierarchical models. *Bayesian Analysis*, 1 :1–19.

[Gkanogiannis and Kalamboukis, 2008] Gkanogiannis, A. and Kalamboukis, T. (2008). A novel supervised learning algorithm and its use for spam detection in social bookmarking systems. In *ECML PKDD Discovery Challenge '08*.

[Goldberger et al., 2003] Goldberger, J., Gordon, S., and Greenspan, H. (2003). An efficient image similarity measure based on approximations of kl-divergence between two gaussian mixtures. In *In Proc. ICCV*, pages 487–493.

[Guha et al., 1998] Guha, S., Rastogi, R., and Shim, K. (1998). Cure : An efficient clustering algorithm for large databases. In Haas, L., Drew, P., Tiwary, A., and Franklin, M., editors, *SIGMOD '98 : Proceedings of the 1998 ACM SIGMOD International Conference on Management of Data*, pages 73–84. ACM Press.

[Gunes and Piccardi, 2006] Gunes, H. and Piccardi, M. (2006). Assessing facial beauty through proportion analysis by image processing and supervised learning. *International Journal of Man-Machine Studies*, 64(12) :1184–1199.

[Haase and Völker, 2005] Haase, P. and Völker, J. (2005). Ontology learning and reasoning - dealing with uncertainty and inconsistency. In *Proceedings of the Workshop on Uncertainty Reasoning for the Semantic Web (URSW)*, pages 45–55.

[Hand and Till, 2001] Hand, D. J. and Till, R. J. (2001). A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning*, 45 :171–186.

[Hartley, 1928] Hartley, R. V. L. (1928). Transmission of information. *Bell Syst. Tech. Journal*, 7 :535–563.

[Harwood et al., 1995] Harwood, D., Ojala, T., Pietikäinen, M., Kelman, S., and Davis, L. S. (1995). Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions. 16 :1–10.

[Hastie and Tibshirani, 1996] Hastie, T. and Tibshirani, R. (1996). Discriminant adaptive nearest neighbor classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 18 :607–616.

[Hastie et al., 2001] Hastie, T., Tibshirani, R., and Friedman, J. (2001). *The Elements of Statistical Learning*. Springer New York Inc., New York, NY, USA.

[Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning : Data Mining, Inference and Prediction*. Springer, second edition.

[Hipp et al., 2001] Hipp, J., Güntzer, U., and Grimmer, U. (2001). Data quality mining – making a virtue of necessity. In *Proceedings of the 6th ACM Sigmod Workshop on Research Issues In Data Mining and Knowledge Discovery (DMKD 2001*, pages 52–57.

[Hoggart et al., 2008] Hoggart, C. J., Whittaker, J. C., De Iorio, M., and Balding, D. J. (2008). Simultaneous analysis of all SNPs in genome-wide and re-sequencing association studies. *PLoS genetics*, 4.

[Howe, 2008] Howe, J. (2008). *Crowdsourcing : Why the Power of the Crowd is Driving the Future of Business*. Crown Business, New York.

[Hsueh et al., 2009] Hsueh, P.-Y., Melville, P., and Sindhwani, V. (2009). Data quality from crowdsourcing : A study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 Workshop on Active Learning for Natural Language Processing*, HLT '09, pages 27–35, Stroudsburg, PA, USA. Association for Computational Linguistics.

[Hubauer et al., 2013] Hubauer, T. M., Lamparter, S., Roshchin, M., Solomakhina, N., and Watson, S. (2013). Analysis of data quality issues in real-world industrial data. In *Proceedings of the 2013 Annual Conference of the Prognostics and Health Management Society*.

[Hui and Zhou., 1998] Hui, S. L. and Zhou., X. H. (1998). Evaluation of diagnostic tests without a gold standard. In Wang, J. Z., Boujemaa, N., Ramirez, N. O., and Natsev, A., editors, *Statistical Methods in Medical Research*, pages 354–370. ACM.

[Imbens and Manski, 2004] Imbens, G. and Manski, C. (2004). Confidence intervals for partially identified parameters. *Econometrica 72*, pages 1845–1857.

[Jamshidian and Jennrich, 1993] Jamshidian, M. and Jennrich, R. I. (1993). Conjugate gradient acceleration of the EM algorithm. *Journal of the American Statistical Association*, 88(421) :221–228.

[Jeffreys, 1946] Jeffreys, H. (1946). An invariant form for the prior probability in estimation problems. In *Proceedings of the Royal Society of London*.

[Jolliffe, 2002] Jolliffe, I. T. (2002). *Principal Component Analysis*. Springer, second edition.

[Jolliffe et al., 2003] Jolliffe, I. T., Trendafilov, N. T., and Uddin, M. (2003). A modified principal component technique based on the LASSO. *Journal of Computational and Graphical Statistics*, 12 :531–547.

[Jordan, 2006] Jordan, M. (2006). An introduction to graphical models. 88.

[Jordan, 1997] Jordan, M. I. (1997). *An Introduction to Graphical Models*. Berkeley, U.C.

[Jordan, 1999] Jordan, M. I. (1999). *Learning in Graphical Models*. MIT Press, Cambridge, MA, USA.

[Kanj, 2013] Kanj, S. (2013). *Méthodes d'Apprentissage pour la Classification Multi Label*. PhD thesis, Université de Technologie de Compiègne.

[Kass, 1980] Kass, G. V. (1980). An exploratory technique for investigating large quantities of categorical data. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 29 :119–127.

[Kass and Wasserman, 1996] Kass, R. E. and Wasserman, L. (1996). The selection of prior distributions by formal rules. *Journal of the American Statistical Association*, 91 :1343–1370.

[Klautau et al., 2002] Klautau, A., Jevtic, N., and Orlitsky, A. (2002). Combined binary classifiers with applications to speech recognition. In *Nearest-neighbor Ecoc With Application to All-pairs Multiclass SVMN*, pages 2469–2472.

[Klein et al., 2002] Klein, D., Kamvar, S. D., and Manning, C. D. (2002). From instance-level constraints to space-level constraints : Making the most of prior knowledge in data clustering. In *Proceedings of the Nineteenth International Conference on Machine Learning*, ICML '02, pages 307–314, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

[Knight, 1921] Knight, F. H. (1921). *Risk, Uncertainty and Profit*. Houghton Mifflin Co, Boston, MA.

[Kotsiantis, 2007] Kotsiantis, S. B. (2007). Supervised machine learning : A review of classification techniques. In *Proceedings of the 2007 Conference on Emerging Artificial Intelligence Applications in Computer Engineering : Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, pages 3–24, Amsterdam, The Netherlands, The Netherlands. IOS Press.

[Lê Cao et al., 2011] Lê Cao, K., Boitard, S., and Besse, P. (2011). Sparse PLS discriminant analysis : Biologically relevant feature selection and graphical displays for multiclass problems. *BMC bioinformatics*, 12(1) :253.

[Lê Cao and Le Gall, 2011] Lê Cao, K. and Le Gall, C. (2011). Integration and variable selection of omics data sets with PLS : a survey. *Journal de la Société Française de Statistique*, 152(2) :77–96.

[Laws et al., 2011] Laws, F., Scheible, C., and Schütze, H. (2011). Active learning with amazon mechanical turk. In *EMNLP*, pages 1546–1556. ACL.

[Lease, 2011] Lease, M. (2011). On quality control and machine learning in crowdsourcing. In *Human Computation*, volume WS-11-11 of *AAAI Workshops*. AAAI.

[Lebart et al., 2000] Lebart, L., Morineau, A., and Piron, M. (2000). *Statistique Exploratoire Multidimensionnelle*.

[Lihong et al., 2009] Lihong, Z., Ying, S., Yushi, Z., Cheng, Z., and Yi, Z. (2009). Face recognition based on multi-class SVM. In *Control and Decision Conference*, pages 5871–5873. IEEE.

[Liu, 2007] Liu, B. (2007). Uncertainty theory. *Springer-Verlag*.

[McLachlan and Krishnan, 2008] McLachlan, G. and Krishnan, T. (2008). *The* EM *algorithm and extensions*. Wiley, Hoboken, NJ, 2 edition.

[Meilijson, 1989] Meilijson, I. (1989). A fast improvement to the ECM algorithm on its own terms. *Journal of the Royal Statistical Society Series*, B-51 :127–138.

[Meng and Rubin, 1993] Meng, X. L. and Rubin, D. B. (1993). Maximum likelihood estimation via the ECM algorithm : A general framework. *Biometrika*, 80(2) :267–278.

[Mitchell, 1997] Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill, New York.

[Molenberghs et al., 2001] Molenberghs, G., Kenward, M., and Goetghebeur, E. (2001). Sensitivity analysis for incomplete contingency tables. *Appl. Statist*, 50 :15–29.

[Molla et al., 2004] Molla, M., Waddell, M., Page, D., and Shavlik, J. W. (2004). Using machine learning to design and interpret gene-expression microarrays. *AI Magazine*, 25 :23–44.

[Morgan and Sonquist, 1963] Morgan, J. N. and Sonquist, J. A. (1963). Problems in the analysis of survey data, and a proposal. *Journal of the American Statistical Association*.

[Neal and Hinton, 1999] Neal, R. and Hinton, G. (1999). A view of the EM algorithm that justifies incremental, sparse, and other variants. In Jordan, M. I., editor, *Learning in Graphical Models*, pages 355–368. MIT Press.

[Neal and Hinton, 1998] Neal, R. and Hinton, G. E. (1998). A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, pages 355–368. Kluwer Academic Publishers.

[Ng and McLachlan, 2003] Ng, S. K. and McLachlan, G. J. (2003). On the choice of the number of blocks with the incremental EM algorithm for the fitting of normal mixtures. *Statistics and Computing*, 13(1) :45–55.

[Nocedal and Wright, 2003] Nocedal, J. and Wright, S. (2003). Numerical optimization (2nd edition). *Springer-Verlag*.

[Nocedal and Wright, 2006] Nocedal, J. and Wright, S. (2006). *Numerical optimization*. Springer, New York, NY, 2. ed. edition.

[Nowak and Rüger, 2010] Nowak, S. and Rüger, S. M. (2010). How reliable are annotations via crowdsourcing : a study about inter-annotator agreement for multi-label image annotation. In Wang, J. Z., Boujemaa, N., Ramirez, N. O., and Natsev, A., editors, *Multimedia Information Retrieval*, pages 557–566. ACM.

[Ormrod, 2012] Ormrod, J. E. (2012). *Human Learning*. Pearson.

[Palmer, 2000] Palmer, T. N. (2000). Predicting uncertainty in forecasts of weather and climate. *Reports on Progress in Physics*, 63(2) :71.

[Paris et al., 2012] Paris, S., Halkias, X., and Glotin, H. (2012). Sparse coding for histograms of local binary patterns applied for image categorization : Toward a bag-of-scenes analysis. In *ICPR*, pages 2817–2820.

[Pearl, 1988] Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[Pearson, 1894] Pearson, K. (1894). Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London*, 185 :71–110.

[Pearson, 1901] Pearson, K. (1901). On lines and planes of closest fit to systems of points in space. *Philosophical Magazine*, 2 :559–572.

[Pereyra et al., 2012] Pereyra, M. A., Dobigeon, N., Batatia, H., and Tourneret, J.-Y. (2012). Segmentation of skin lesions in 2d and 3d ultrasound images using a spatially coherent generalized rayleigh mixture model. *IEEE Transactions on Medical Imaging*, 31(8) :1509–1520.

[Pipino et al., 2002] Pipino, L. L., Lee, Y. W., and Wang, R. Y. (2002). Data quality assessment. *Commun. ACM*, 45 :211–218.

[Platt, 1999] Platt, J. C. (1999). Advances in kernel methods. chapter Fast Training of Support Vector Machines Using Sequential Minimal Optimization, pages 185–208. MIT Press, Cambridge, MA, USA.

[Pontil and Verri, 1998] Pontil, M. and Verri, A. (1998). Properties of support vector machines. *Neural Computation*, 10 :955–974.

[Quafafou, 1997] Quafafou, M. (1997). Learning flexible concepts from uncertain data. *10th International Symposium, ISMIS'97*, pages 507–518.

[Raykar and Yu, 2011] Raykar, V. C. and Yu, S. (2011). Ranking annotators for crowdsourced labeling tasks. In *NIPS*, pages 1809–1817.

[Raykar and Yu, 2012] Raykar, V. C. and Yu, S. (2012). Eliminating spammers and ranking annotators for crowdsourced labeling tasks. *J. Mach. Learn. Res.*, 13 :491–518.

[Raykar et al., 2010] Raykar, V. C., Yu, S., Zhao, L. H., and Valadez, G. H. (2010). Learning from crowds. *In Journal of Machine Learning Research 11 - MIT Press*, pages 1297–1322.

[Robert, 2007] Robert, C. (2007). *The Bayesian Choice, From Decision-Theoretic Foundations to Computational Implementation*. Springer-Verlag.

[Sallis et al., 2011] Sallis, P. J., Claster, W., and Hernández, S. (2011). A machine-learning algorithm for wind gust prediction. *Computers and Geosciences*, 37(9) :1337–1344.

[Saporta, 2011] Saporta, G. (2011). *Probabilités, analyse des données et statistique*. Éd. Technip, Paris.

[Sarasua et al., 2012] Sarasua, C., Simperl, E., and Noy, N. F. (2012). CrowdMAP : Crowdsourcing ontology alignment with microtasks. In *Proceedings of the 11th International Semantic Web Conference (ISWC2012)*, pages 525–541.

[Schapire and Singer, 2000] Schapire, R. E. and Singer, Y. (2000). Boostexter : A boosting-based system for text categorization. *Machine Learning*, 39 :135–168.

[Shafer, 1976] Shafer, G. (1976). *A Mathematical Theory of Evidence*. Princeton University Press.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell System Technical Journal*, 27 :379–423.

[Shannon and Weaver, 1949] Shannon, C. E. and Weaver, W. (1949). *The Mathematical Theory of Communication*. University of Illinois Press, Urbana, Chicago, and London.

[Shen and Huang, 2008] Shen, H. and Huang, J. Z. (2008). Sparse principal component analysis via regularized low rank matrix approximation. *J. Multivar. Anal.*, 99(6) :1015–1034.

[Sheng et al., 2008] Sheng, V. S., Provost, F., and Ipeirotis, P. G. (2008). Get another label ? improving data quality and data mining using multiple, noisy labelers. In *Proceedings of the 14th International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 614–622, New York, NY, USA. ACM.

[Smyth et al., 1995] Smyth, P., Fayyad, U., Burl, M., Perona, P., and Baldi, P. (1995). Inferring ground truth from subjective labelling of venus images. *NIPS*, pages 1085–1092.

[Snow et al., 2008] Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good ? evaluating non-expert annotations for natural language tasks. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 254–263.

[Sorokin and Forsyth, 2008] Sorokin, A. and Forsyth, D. (2008). Utility data annotation with amazon mechanical turk. *CVPRW '08. IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–8.

[Tenenhaus, 1998] Tenenhaus, M. (1998). *La régression* PLS *: Théorie et Pratique*. Éd. Technip, 1998 (05-Gap), Paris.

[Ting, 2000] Ting, K. M. (2000). A comparative study of cost-sensitive boosting algorithms. In *Proceedings of the 17th International Conference on Machine Learning*, pages 983–990. Morgan Kaufmann.

[van Asselt et al., 2002] van Asselt, M. B. A., Rotmans, J., and Bird (2002). Uncertainty in integrated assessment modelling : from positivism to pluralism. *Climatic Change*, 54 :75–105.

[Vansteelandt et al., 2006] Vansteelandt, S., Goetghebeur, E., Kenward, G., and Molenberghs, G. (2006). Ignorance and uncertainty regions as inferential tools in a sensitivity analysis. *Statistica Sinica 16*, pages 953–979.

[Vapnik, 1995] Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, NY, USA.

[Vapnik, 1998] Vapnik, V. N. (1998). *Statistical Learning Theory*. Wiley, 1 edition.

[von Ahn et al., 2004] von Ahn, L., Blum, M., and Langford, J. (2004). Telling humans and computers apart automatically. *Commun. ACM*, 47 :56–60.

[von Ahn and Dabbish, 2004] von Ahn, L. and Dabbish, L. (2004). Labeling images with a computer game. In *CHI '04 : Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, New York, NY, USA. ACM.

[von Ahn et al., 2008] von Ahn, L., Maurer, B., Mcmillen, C., Abraham, D., and Blum, M. (2008). reCAPTCHA : Human-based character recognition via web security measures. *Science*, 321 :1465–1468.

[Walker et al., 2003] Walker, W. E., Harremoës, P., Rotmans, J., van der Sluijs, J. P., van Asselt, M. B. A., Janssen, P., and von Krauss, M. P. K. (2003). Defining uncertainty : A conceptual basis for uncertainty management in model-based decision support. *Integrated Assessment*, 4 :5–17.

[Wasserman, 2004] Wasserman, L. (2004). *All of Statistics : A Concise Course in Statistical Inference*. Springer.

[Wazaefi, 2013] Wazaefi, Y. (2013). *Automatic Diagnosis of Melanoma from Dermoscopic Images of Melanocytic Tumors : Analytical and Comparative Approaches*. PhD thesis, Université d'Aix-Marseille.

[Welinder and Perona, 2010] Welinder, P. and Perona, P. (2010). Online crowdsourcing : Rating annotators and obtaining cost-effective labels. In *W. on Advancing Computer Vision with Humans in the Loop*, pages 1526–1534.

[West et al., 2001] West, M., Blanchette, C., Dressman, H., Huang, E., Ishida, S., Spang, R., Zuzan, H., Olson, J. A., Marks, J. R., and Nevins, J. R. (2001). Predicting the clinical status of human breast cancer by using gene expression profiles. *Proc. Natl. Acad. Sci. U.S.A.*, 98.

[Whitehill et al., 2009] Whitehill, J., Ruvolo, P., Wu, T., Bergsma, J., and Movellan, J. R. (2009). Whose vote should count more : Optimal integration of labels from labelers of unknown expertise. In *NIPS*, pages 2035–2043. Curran Associates, Inc.

[Whiteside, 1974] Whiteside, D. T. (1974). The mathematical papers of Isaac Newton. *Cambridge University Press*, I-VII.

[Wolley and Quafafou, 2012a] Wolley, C. and Quafafou, M. (2012a). Learning from multiple annotators : when data is hard and annotators are unreliable. *12th International Conference on Data Mining Workshops (ICDMW)*, pages 514–521.

[Wolley and Quafafou, 2012b] Wolley, C. and Quafafou, M. (2012b). Learning from multiple naive annotators. *8th International Conference on Advanced Data Mining and Applications (ADMA)*, 7713 :173–185.

[Wolley and Quafafou, 2013a] Wolley, C. and Quafafou, M. (2013a). Multiclass learning from multiple uncertain annotations. *12th International Symposium on Advances in Intelligent Data Analysis (IDA)*, 8207 :438–449.

[Wolley and Quafafou, 2013b] Wolley, C. and Quafafou, M. (2013b). Scalable experts selection when learning from noisy labelers. *12th International Conference on Machine Learning and Applications (ICMLA) Poster Session*.

[Wu, 1983] Wu, C. F. J. (1983). On the convergence properties of the EM algorithm. *The Annals of Statistics*, 11(1) :95–103.

[Xu and Jordan, 1995] Xu, L. and Jordan, M. I. (1995). On convergence properties of the EM algorithm for gaussian mixtures. *Neural Computation*, 8 :129–151.

[Yan et al., 2010] Yan, Y., Hermosillo, G., Rosales, R., Bogoni, L., Fung, G., Moy, L., Schmidt, M., and Dy, J. (2010). Modeling annotator expertise : Learning when everybody knows a bit of something. *In Proceedings of the 13th International Conference on Artificial Intelligence and Statistics (AISTATS)*.

[Yan et al., 2012] Yan, Y., Rosales, R., Fung, G., Farooq, F., Rao, B., and Dy, J. G. (2012). Active learning from multiple knowledge sources. In Lawrence, N. D. and Girolami, M., editors, *AISTATS*, volume 22, pages 1350–1357.

[Zadeh, 1978] Zadeh, L. (1978). Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1 :3–28.

[Zadeh, 1965] Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, 8 :338–353.

[Zhang and Obradovic, 2012] Zhang, P. and Obradovic, Z. (2012). Integration of multiple annotators by aggregating experts and filtering novices. In *BIBM'12*, pages 1–6.

[Zhang, 2003] Zhang, Z. (2003). Learning metrics via discriminant kernels and multidimensional scaling : Toward expected euclidean representation. In Fawcett, T. and Mishra, N., editors, *ICML*, pages 872–879. AAAI Press.

[Zighed et al., 2010] Zighed, D. A., Ritschard, G., and Marcellin, S. (2010). Asymmetric and sample size sensitive entropy measures for supervised learning. In *Advances in Intelligent Information Systems*, volume 265, pages 27–42. Springer.

[Zou et al., 2004] Zou, H., Hastie, T., and Tibshirani, R. (2004). Sparse principal component analysis. *Journal of Computational and Graphical Statistics*, 15 :2006.