Semi-supervised learning in insurance : fairness and active learning

François Hu

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2022IPPAG002

Thèse de doctorat

ENSAE

IP PARIS

Semi-supervised learning in insurance: fairness and active learning

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée à l'École nationale de la statistique et de l'administration économique

École doctorale n°574 École Doctorale de Mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Palaiseau, le 15 juin 2022, par

FRANÇOIS HU

Composition du Jury :

Christian-Yann ROBERT Professeur, École nationale de la statistique et de l'administration économique (CREST)	Président
Christophe DUTANG Maître de conférences, Université Paris Dauphine (CEREMADE)	Rapporteur
Olivier WINTENBERGER Professeur, Université Pierre et Marie Curie (LPSM)	Rapporteur
Arthur CHARPENTIER Professeur, Université du Québec à Montréal (QUANTACT)	Examinateur
Stéphane LOISEL Professeur, Université Lyon 1 (SAF)	Examinateur
Caroline HILLAIRET Professeure, École nationale de la statistique et de l'administration économique (CREST)	Directrice de thèse
Romuald ELIE Professeur, Université Gustave Eiffel (LAMA)	Directeur de thèse (Invité)
Marc JUILLARD Directeur du DataLab, Société Générale Assurances	Invité

Dedication and acknowledgements

Tout d'abord, je tiens à remercier chaleureusement mes encadrants de thèse Caroline Hillairet, Romuald Elie et Marc Juillard. Ils se sont bien complétés tout au long de cette thèse. Ils m'ont fait confiance et m'ont beaucoup soutenu durant toute cette aventure.

Je tiens à remercier sincèrement Christophe Dutang et Olivier Wintenberger qui, en dépit de leur emploi du temps chargé, ont accepté de rapporter cette thèse. Je suis également très reconnaissant à Stéphane Loisel, Arthur Charpentier et Christian-Yann Robert pour l'intérêt qu'ils ont porté à mes recherches en acceptant d'être membres de mon jury de thèse.

Ma gratitude va également à toute l'équipe du DataLab de la Société Générale Assurances. Je tiens à les remercier pour leur soutien et leurs conseils avisés. Les discussions scientifiques et politiques menées dans la bonne humeur pendant les pauses café et les petits déjeuners ont été très enrichissantes.

Mes remerciements vont également au laboratoire Finance-Assurances du CREST-ENSAE. Je les remercie de m'avoir immergé dans un environnement scientifique très stimulant.

Je remercie ma famille et mes amis de m'avoir soutenue tout au long de ma scolarité et de m'avoir rappelé qu'il est bon de faire une pause de temps en temps.

Je tiens particulièrement à remercier ma copine qui m'a toujours remonté le moral dans les moments difficiles.

Table of Contents

Chapter 1

Introduction générale

Contents

Cette thèse aborde et propose des solutions concrètes aux problèmes liés à la performance des modèles d'apprentissage statistique dans le domaine actuariel, que ce soit concernant le taux de bonnes prédictions (*accuracy* en anglais) ou de son équité (*fairness* en anglais). Cette thèse est une thèse CIFRE résultant d'une recherche partenariale entre le Laboratoire de Finance et Assurance (LFA) du CREST, ENSAE - Institut Polytechnique de Paris, et le service datalab de la Société Générale Assurances. Dans ce chapitre introductif, nous décrivons le contexte de notre étude et formulons formellement le problème avant de faire une revue de la littérature à deux thèmes centraux de cette thèse : l'apprentissage actif et l'équité algorithmique. Enfin, nous résumons nos contributions.

1.1 Contexte et problématique

1.1.1 Données et modèles d'apprentissage

Aujourd'hui de nombreuses entreprises font face à une masse importante de données brutes. Ces données peuvent être de natures différentes.

- D'un côté, nous avons des données structurées qui, comme son nom l'indique, sont des données organisées et soigneusement formatées. Les données structurées peuvent être des enregistrements (ou des transactions) dans un environnement de base de données relationnelles SQL, constituées de tables avec des champs prédéfinis ;

- D'un autre côté, nous avons des données non structurées qui ne sont ni organisées ni formatées. Les exemples de données non structurées les plus courants sont les fichiers texte, les images, les fichiers vidéo et les fichiers audio.

Cette explosion en matière de données est accompagnée par de nombreuses avancées technologiques sophistiquées permettant de les exploiter: nous assistons au cours du dernier siècle à la démocratisation des solutions d'intelligence artificielle (IA) basées sur l'apprentissage automatique (*machine learning* a.k.a. *ML*). Nous pouvons par exemple citer la mise en place des voitures autonomes, des traducteurs automatiques et des programmes informatique permettant de jouer aux jeux de plateau et de gagner contre les meilleurs joueurs professionnels (e.g. échec, jeu de go, ...). Toutefois, pour obtenir ces solutions IA, il est classiquement nécessaire d'avoir au préalable des données suffisantes (voire massives !), de bonnes qualités, non biaisées et étiquetées.

Données massives. Les systèmes d'apprentissage automatique sont capables d'apprendre des données et de s'adapter au fil du temps sans suivre d'instructions spécifiques ou de code programmé[1]. Les algorithmes récents (exemple : les méthodes d'apprentissage profond a.k.a. *Deep Learning*) sont de plus en plus *complexes* (car de plus en plus de *paramètres* à calibrer) et ainsi deviennent très gourmands en termes de données [NVK+15, ZWLD16, MWW+18]: pour un système complexe, plus il reçoit de données (sur lesquels le système *apprend*) plus il est potentiellement performant pour faire des analyses prédictives.

Données de bonne qualité et fiables. Les solutions IA peuvent être très performantes si les données utilisées pour les *entraîner* sont de bonne qualité, entre autres, des données non-erronées, complètes et précises. En revanche, si les données ne sont pas assez qualitatives alors nous risquerons un effet "*Garbage-In-Garbage-Out*" (GIGO) [KHE+16, SS17, GYY+20]: si les données d'entrée défectueues ou absurdes alors le résultat d'un algorithme est également défectueux ou absurde. De plus, via ses prédictions, les algorithmes IA alimentés par des données défectueuses peuvent amplifier ou propager des erreurs dans les données. Parmi les exemples, citons les algorithmes d'apprentissage semi-supervisé qui ne nécessitent pas l'intervention d'un expert humain pour étiqueter les données mais la prédiction de modèles d'apprentissage (la notion d'apprentissage semi-supervisé et ses impacts seront détaillés dans la section 1.2.2). L'efficacité des capacités d'apprentissage d'un algorithme d'apprentissage automatique dépend non seulement de la quantité de donnée fournie mais aussi la qualité de celles-ci et du degré d'"*informativité*" et de la "*représentativité*" qu'elles contiennent.

Données non biaisées. Les solutions IA sont en constante évolution et leur manque d'interprétabilité peut poser problème: les entreprises doivent s'assurer que leur solution IA ne discrimine pas et n'induit pas de biais involontaires, ce qui peut les exposer à des risques opérationnels et de réputation. Les études existantes sur l'équité dans la prise de décision automatisée critiquent l'apprentissage automatique, sans contrainte d'équité, de discriminer les groupes historiquement sous-représentés ou défavorisés dans la population [otPMD+16, BS16].

Données étiquetées. Les données (non étiquetées) sont aujourd'hui disponibles en grande quantité tout autour de nous : les e-mails dans notre compte, des centaines de photos dans notre téléphone, des dizaines de vidéos de présentation et d'enregistrements de discours... Cependant la majorité des algorithmes utilisés aujourd'hui ont besoin de données étiquetées pour *apprendre*, de sorte que le traitement des données brutes dont nous disposons reste l'alternative la plus réalisable pour former des modèles d'apprentissage automatique. Les données étiquetées sont très précieuses car elles indiquent (explicitement) à la machine ce qu'elle doit rechercher. Cela facilite la construction de modèles de prédictions complexes. Une fois l'algorithme d'apprentissage automatique entraîné, il est capable de

[1] Dans le passé les entreprises construisaient des systèmes basés sur des règles (métiers) essayant d'englober un large éventail d'utilisations analytiques. Mais ces derniers sont souvent fragiles et incapables de gérer des besoins commerciaux en constante évolution.

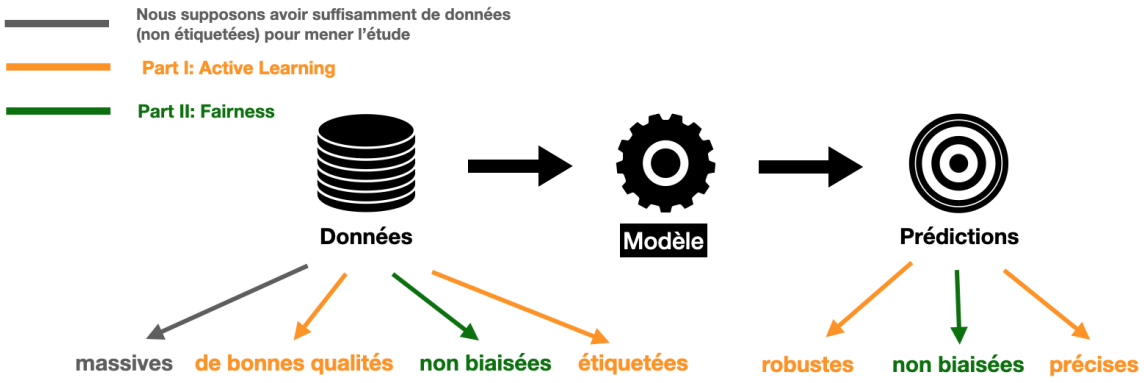


Figure 1.1: Processus d'apprentissage automatique et domaines abordés dans cette thèse. Schéma classique des propriétés nécessaire (à savoir des données massives, de bonnes qualités, non biaisées et étiquetées) pour avoir un modèle ML performant (des prédictions robuste, non biaisées et précises).

tirer des inférences dans les nouveaux ensembles de données. Les étiquettes peuvent être obtenues en demandant aux humains (ou *Oracle*) de porter des jugements sur un élément donné de données non étiquetées. Il est à noter que les données étiquetées sont souvent beaucoup plus coûteuses à obtenir que les données brutes non étiquetées [NVK+15, SSD18].

Via le déploiement des modèles, l'apprentissage automatique apporte un atout considérable dans des processus d'automatisation comme la catégorisation des documents textuels [GR08, HKS17, ZSAT19, DMDR21, ARN+21] ou la détection d'objets dans les images [MPH+17, MFA+18, ZWY+20] ou dans les vidéos [BB20, BAT+21]. En assurance, un exemple concret serait d'extraire des informations de documents numérisés tels que les actes de décès (en *assurance vie*) ou des photos de voitures endommagées (en *assurance dommages*) avant de les catégoriser. Un autre exemple concret serait de détecter automatiquement les documents textuels qui ne sont pas conformes à la norme RGPD. En marketing, un exemple serait de détecter automatiquement à partir des avis clients recueillis les motifs d'insatisfaction des clients pour les améliorer. Néanmoins, en pratique les entreprises font face à de nombreux défis majeurs, parmi eux :
(1) la construction des modèles d'apprentissage *performants* avec un **budget d'annotation restreint**
(2) l'équité des modèles d'apprentissage en présence de **données biaisées**.
Le schéma 1.1 donne un aperçu (très simpliste) des conditions classiques à avoir dans les données afin d'obtenir un modèle prédictif performant. En pratique, que fait-on si une ou plusieurs de ces conditions ne sont pas vérifiées ? Cette thèse, composée de deux parties (Active Learning et Fairness) tente de répondre à cette question.

1.1.2 Annotation active des données : améliorer la performance des modèles

L'un des défis majeurs en IA est l'acquisition des données étiquetées [KGK15, JM15, WCW+17, KEUR+18, RHW19, GYY+20]. Dans le domaine de l'apprentissage automatique, l'étiquetage des données est le processus d'identification des données brutes (images, fichiers texte, vidéos, etc.) et l'ajout d'une ou plusieurs étiquettes significatives et informatives pour fournir un contexte afin qu'un modèle d'apprentissage automatique puisse en tirer des enseignements. Ainsi, la précision des modèles d'apprentissage s'améliore considérablement si elle a été entraînée sur des données étiquetées. Et aujourd'hui, dans la plupart des cas, ces données doivent être étiquetées par des humains. Par exemple, en assurance santé, la détection automatique de cas frauduleux et abusifs via un modèle ML nécessite l'intervention des experts du domaine pour étiqueter les données d'entraînement [KGK15]. De ce fait plusieurs entreprises investissent dans l'étiquetage des données et plusieurs plateformes spécialisées dans l'étiquetage des données ont vu le jour récemment. Nous pouvons citer par exemple *Labelbox*[2] un outil facilitant l'annotation et *Amazon SageMaker Data Labeling*[3] l'une des principales plateformes offrant une main-d'œuvre d'étiquetage à la demande.

Souvent en pratique, contrairement aux données structurées, les données non structurées (les textes par exemple) ne sont pas étiquetées et l'étiquetage de ces dernières pose un problème de coût et de temps. En effet, un étiquetage exige l'intervention d'un expert humain rigoureux (d'où le *coût*) qui doit analyser et étiqueter les données une à une (d'où le *temps*), notamment lorsque les données sont sous forme de son, de vidéo ou de texte. Citons l'exemple suivant : la détection de la conformité juridique des documents textuels (un exemple de conformité est la conformité GDPR[4]). Dans cet exemple l'intervention d'un juriste est indispensable compte-tenu des connaissances juridiques nécessaires pour décider de la conformité de ces documents textuels. Effectivement, à titre d'exemple à la Société Générale Assurances, des sources de données textuelles volumineuses et potentiellement à risque sont les zones de texte libre utilisées par les télé-conseillers qui contiennent plus d'un million de verbatims par an. Cependant s'il est possible d'étiqueter, par des experts humains, les données échantillonnées aléatoirement, ce processus d'échantillonnage (que nous appelons *passif*) n'est souvent pas adapté à la gestion de grand volume. Face à la gestion quasi impossible de cette masse importante de données et à la rareté des documents non conformes, la construction classique d'une base d'entraînement, à savoir l'étiquetage aléatoire un à un des documents, a été vite abandonnée[5].

L'idée serait donc d'annoter d'une manière guidée en se focalisant sur l'annotation des données *utiles*. De manière naïve, nous voulons éviter d'étiqueter les données redondantes, non informatives, non représentatives ou celles qui contiennent des erreurs. Pour ce faire, il serait possible de construire un modèle statistique sélectionnant les données les plus informatives à étiqueter : au lieu d'un étiquetage

[2] https://labelbox.com/

[3] https://aws.amazon.com/fr/sagemaker/data-labeling/

[4] **GDPR** : Règlement général sur la protection des données, un règlement européen qui constitue le texte de référence en matière de protection des données à caractère personnel.

[5] Les juristes ne peuvent étiqueter que quelques milliers de documents par an et le ratio de non-conformité est inférieur a priori à 1/1000

aléatoire des données, nous nous concentrons sur un étiquetage "intelligent" des données. Dans la littérature ce processus d'apprentissage s'appelle l'apprentissage actif (ou *active learning* en anglais). De manière plus formelle, il correspond au processus d'annotation et d'entraînement du modèle d'apprentissage qui a pour objectif de maximiser la performance tout en minimisant la quantité de données annotées (en l'occurence nous choisissons les données les plus *qualitatives* selon un modèle d'apprentissage donné). Trois composantes sont nécessaires pour ce processus:
(1) avoir un modèle d'apprentissage préalablement entraîné (nous supposons que nous avons des données étiquetées de départ);
(2) avoir des données non étiquetées massives;
(3) avoir à disposition un ou plusieurs experts humain pour l'étiquetage.
Il est à noter que la construction d'un modèle d'apprentissage à l'aide à la fois des données étiquetées et des données non étiquetées s'appelle l'*apprentissage semi-supervisé* (ou *semi-supervised learning* en anglais). Ce dernier sera introduit dans la sous-section 1.2.2.

1.1.3 Équité algorithmique : limiter les biais algorithmiques

Un autre défi majeur en IA est l'impact des données biaisées sur les modèles d'apprentissage statistique [Cho17, SGCA18, JN20, MMS+21, ADS+22]. Les algorithmes d'apprentissage statistique apprennent sur des données qui peuvent être biaisées, soulevant en conséquence des problèmes de précision ou d'équité (*fairness* en anglais). Ces dernières années, avec l'essor de la data science, un domaine récent de recherche se développe: l'*équité algorithmique* (*Algorithmic fairness*). L'équité algorithmique a pour objectif de réduire les prédictions discriminatoires basées, par exemple, sur des attributs sensibles (ou attributs protégés) tels que le genre ou l'orientation sexuelle. Cette réduction évite ainsi que les biais dans les données ne soient propagés et amplifiés par les méthodes ML.

La raison de l'inéquité se trouve directement dans les données. En effet, Les systèmes d'IA apprennent à prendre des décisions sur la base de données étiquetées. Ces données peuvent

- (*Biais de mesure*) inclure des décisions humaines biaisées (par exemple, l'algorithme de recrutement qui favorise les candidats masculins[6]);
- (*Préjugés sociaux*) refléter des inégalités sociales (par exemple, le logiciel de justice COMPAS qui catégorise le niveau de risque de criminalité en fonction de l'origine ethnique des individus [7]);
- (*Biais de représentation*) être tout simplement trop déséquilibrées pour être représentatives, ce qui entraîne des erreurs importantes pour les classes minoritaires (par exemple, le cas de la reconnaissance du genre chez Microsoft[8]).

[6] https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

[7] https://www.nouvelobs.com/rue89/rue89-etats-unis/20160524.RUE2964/etats-unis-un-algorithme-qui-predit-les-recidives-lese-les-noirs.html

[8] https://www.theverge.com/2019/1/25/18197137/amazon-rekognition-facial-recognition-bias-race-gender

Nous renvoyons à [MMS+21] pour une discussion générale sur les principaux types de préjugés et de discrimination dans le cadre de l'apprentissage automatique.

Pour les assureurs de l'UE, le principe d'équité est une obligation: il est reconnu dans la directive IDD (pour Insurance Distribution Directive, une directive européenne qui fixe les règles relatives à la distribution d'assurances) que les distributeurs d'assurance doivent "toujours agir de manière honnête, équitable et professionnelle, conformément aux meilleurs intérêts de leurs clients" (Art. 17(1) - IDD). Le principe d'équité est également énoncé à l'article 5 du *Règlement général sur la protection des données* (RGPD) : les données à caractère personnel doivent être "traitées de manière licite, loyale et transparente à l'égard de la personne concernée" (Art. 5(1) - RGPD). Depuis la décision de la Cour de justice européenne du 21 décembre 2012, les assureurs de l'UE ne doivent plus utiliser le critère du genre dans le calcul des primes d'assurance. Pour les lecteurs intéressés, nous renvoyons à [BC22] pour une discussion générale sur l'équité en assurance, y compris l'impact des contextes historiques et culturels.

Généralement, dans le domaine de l'équité algorithmique, il existe deux classes de mesures d'équité pour mesurer l'injustice des modèles ML : les mesures d'équité individuelles [DHP+12] (*Individual fairness metrics*) et les mesures d'équité de groupe (*Group fairness metrics*).

- L'**équité individuelle** part du principe que les individus similaires doivent recevoir des décisions similaires. Ce principe traite de la comparaison d'individus plutôt que de se concentrer sur des groupes de personnes partageant certaines caractéristiques.

- L'**équité de groupe** part de l'idée qu'il existe des groupes de personnes susceptibles d'être victimes de préjugés et de décisions injustes, et tente donc de parvenir à une égalité de traitement pour les groupes plutôt que pour les individus. La plupart de la littérature existante sur l'équité dans l'apprentissage automatique s'y réfère.

Cette thèse étudie plus particulièrement l'équité de groupe, qui dérive d'un concept de non-discrimination sur la base de l'appartenance à un groupe protégé tel que le genre ou l'âge,

D'un point de vue algorithmique, quitte à réduire la capacité prédictive, assurer l'équité revient généralement à intégrer des contraintes (1) directement dans les données (méthode *pre-processing*) (2) dans le programme d'optimisation permettant d'apprendre une règle de décision à partir des données (méthode *in-processing*) ou (3) dans les prédictions des modèles d'apprentissage (méthode *post-processing*). Il existe plusieurs notions d'équité. Les plus connues sont les suivantes : parité démographique, l'égalité des opportunités et l'égalité des chances.

Parité démographique. la parité démographique (*Demographic parity* en anglais) ou parité de classification est l'une des notions d'équité les plus communément admises. Elle exige que la prédiction soit statistiquement indépendante de l'attribut sensible. En d'autres termes, les taux d'acceptation prévus pour les groupes protégés et non protégés devraient être égaux. Cela signifie par exemple que les entreprises ne doivent pas embaucher proportionnellement plus de candidats d'un groupe que de

l'autre (l'algorithme de recrutement d'Amazon, alimenté par l'IA, peut favoriser implicitement les candidats masculins[9]). En France, des quotas dans les postes de direction des grandes entreprises sont instaurés par la loi du 24 décembre 2021 (n° 2021-1774), avec un objectif de 40% de femmes cadres dirigeantes d'ici à 2030, sous peine de pénalité financière pour les entreprises. La notion de parité démographique semble adaptée lorsque l'étiquette n'est pas fiable en raison des décisions humaines biaisées ou des préjugés sociaux. Un problème de ce type a été observé dans l'outil de prédiction du risque de récidive COMPAS utilisé par les Etats-Unis. Avec cet outil (utilisé pour évaluer le niveau de criminalité), le nombre d'arrestations des groupes minoritaires (d'originie ethnique différente) est significativement plus élevé que celui du reste de la population [ALMK16, DMB16]. Bien que les données utilisées par COMPAS ne comprennent pas explicitement l'origine ethnique d'un individu, d'autres aspects des données y sont corrélés, ce qui conduit à des disparités raciales dans les prédictions.

Égalité des chances. L'égalité des chances (*Equalized odds* en anglais) tient compte à la fois des résultats prédits et des résultats réels. Ainsi, étant donné le résultat réel, la prédiction est conditionnellement indépendante de l'attribut protégé. En d'autres termes, les deux sous-populations doivent avoir le même taux de vrais positifs et taux de faux positifs. Cette notion est adaptée aux scénarios où les étiquettes sont considérées fiables comme par exemple la prédiction de maladies. Elle convient également lorsque l'accent est mis sur le *rappel* (la fraction du nombre total d'instances positives qui sont correctement prédites positives) plutôt que sur la *précision* (s'assurer qu'une instance positive prédite est réellement une instance positive). Par exemple l'équité consisterait à avoir un même taux de faux positifs d'un logiciel de reconnaissance faciale pour tous les groupes ethniques. Un problème de ce type a été observé avec la reconnaissance du genre de Microsoft, qui reconnaît plus souvent le genre des hommes blancs[10].

Égalité des opportunités. L'égalité des opportunités (*Equal opportunity* en anglais) est une notion d'équité issue d'un assouplissement des contraintes de l'égalité des chances : dans le cadre de classification binaire, l'égalité des chances exige la non-discrimination uniquement au sein du groupe de résultats "favorisé" (par exemple, les admissions à l'université ou le recrutement d'employés). Dans l'exemple du recrutement, cela revient à dire que les entreprises devraient embaucher une proportion égale de personnes parmi la fraction qualifiée de chaque groupe.

Ces concepts d'équité seront définis de manière formelle dans la sous-section 1.4.

[9] https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

[10] https://www.newscientist.com/article/2161028-face-recognition-software-is-perfect-if-youre-a-white-man/

1.2 Formulation du problème

Traditionnellement en apprentissage statistique, nous distinguons généralement deux types de tâches d'apprentissage : l'apprentissage non supervisé et l'apprentissage supervisé (resp. *unsupervised learning* et *supervised learning* en anglais). L'apprentissage non supervisé utilise un ensemble de données non étiquetées pour estimer la structure de l'espace des instances. L'apprentissage supervisé consiste à apprendre une fonction de prédiction à partir d'exemples étiquetés.

1.2.1 Apprentissage supervisé "traditionnel"

Nous considérons $\mathcal{X}$ l'espace des instances et $\mathcal{Y}$ l'espace des étiquettes (ou labels ou encore classes). Notons les mesures de probabilité adéquates pour ces espaces : $\mathbb{P}$ la distribution sur $\mathcal{X} \times \mathcal{Y}$ et $\mathbb{P}_{\mathcal{X}}$ la distribution marginale de $\mathbb{P}$ sur $\mathcal{X}$. Posons de plus $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ l'espace des hypothèses (aussi appelé l'ensemble des prédicteurs). Ainsi pour une instance étiquetée $(x, y) \in \mathcal{X} \times \mathcal{Y}$ donnée et pour un prédicteur $h \in \mathcal{H}$ donné, $h(x)$ vise à prédire l'étiquette y de x.

Afin d'évaluer les prédicteurs, une fonction de perte, que nous notons $l : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty[$, doit être définie pour comparer l'étiquette donnée par un prédicteur à la vraie étiquette. Par exemple, nous avons la *perte binaire 0-1* définie par $l_{0-1}(y, y') = \mathbb{1}(y \neq y')$. Cette fonction de perte est plus adaptée pour un problème de classification, c'est-à-dire un ensemble fini de classes $\mathcal{Y}$. Un exemple de fonction de perte adaptée pour la régression ($\mathcal{Y} = \mathbb{R}$) est la *perte quadratique* définie par $l_2(y, y') = (y - y')^2$. Il convient de noter qu'une fonction de perte satisfait souvent les propriétés suivantes, $l(y, y) = 0$ et $l(a, y)$ est une fonction qui est croissante avec la distance entre x et a.

Nous rappelons que la fonction de perte permet d'évaluer la performance d'un prédicteur sur un seul exemple. Pour évaluer les prédicteurs sur un ensemble $\mathcal{X} \times \mathcal{Y}$ une notion du risque doit être définie. Nous appelons *risque théorique*, l'espérance de la perte sous la distribution $\mathbb{P}$. Autrement dit, pour tout $h \in \mathcal{H}$ prédicteur et $l : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty[$ fonction de perte, le risque théorique est défini par

$$R(h) := \mathbb{E}[l(h(x), y)] = \int_{\mathcal{X} \times \mathcal{Y}} l(h(x), y) \mathbb{P}(dx, dy). \tag{1.1}$$

En pratique, la distribution $\mathbb{P}$ étant souvent inconnue, ce risque n'est pas calculable. Nous devons donc l'estimer et un estimateur (naturel) de ce risque est le *risque empirique* qui correspond à la moyenne empirique de la perte pour chaque observation $(x, y) \in \mathcal{X} \times \mathcal{Y}$. Si nous notons $(x_i, y_i)_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ les observations alors le risque empirique sur ces données est formellement défini ci-après

$$\hat{R}(h) := \frac{1}{N} \sum_{i=1}^{N} l(h(x_i), y_i). \tag{1.2}$$

Ces quantités étant ainsi définies nous remarquons qu'en classification, en considérant la perte 0-1,

- le risque théorique, aussi appelé erreur (théorique) de classification, correspond à la probabilité que le prédicteur h prédise une réponse différente de celle de l'oracle :

$$R(h) = \mathbb{E}[\mathbb{1}_{h(x) \neq y}] = \mathbb{P}(h(x) \neq y).$$

- le risque empirique, aussi appelé erreur empirique de classification, s'écrit

$$\hat{R}(h) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{1}_{h(x_i)\neq y_i}.$$

Soit $\mathscr{D}^{(train)} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^{L}$ l'ensemble d'apprentissage et $\mathscr{D}^{(test)} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^{T}$ l'ensemble de test où (x_i, y_i) sont tirés i. i.d. selon la distribution $\mathbb{P}$. Si nous supposons que nous avons accès à un grand ensemble de données non étiquetées désigné par $\mathscr{D}_{\mathscr{X}}^{(pool)} = \{x_1^{(pool)}, \ldots, x_U^{(pool)}\}$ (nous l'appellerons également *pool-set*) ainsi qu'à un ou plusieurs oracles pour étiqueter les données, alors le processus d'apprentissage statistique consiste en les étapes suivantes :

(i) **Phase d'étiquetage.** Cette étape consiste à étiqueter les données "brutes"[11] permettant de constituer des données étiquetées (par exemple à la fois des données d'entraînement $\mathscr{D}^{(train)}$ et des données de test $\mathscr{D}^{(test)}$). Nous considérons deux types d'étiquetage : un *étiquetage passif* qui consiste à requêter l'étiquette des données échantillonnées aléatoirement et un *étiquetage actif* qui requête les données selon un critère d'importance. On note que l'étiquetage passif permet de générer une base de données i.i.d. alors que l'étiquetage actif génère une base de données qui ne vérifie pas la condition d'indépendance. Les données de test doivent être générées avec un étiquetage passif afin de reproduire empiriquement $\mathscr{X} \times \mathscr{Y}$, c'est-à-dire contraindre la base de test à avoir la même distribution que $\mathscr{X} \times \mathscr{Y}$.

(ii) **Phase d'apprentissage**. Étant donné l'ensemble d'apprentissage $\mathscr{D}^{(train)}$, cette étape consiste à trouver un estimateur $\hat{h} \in \mathscr{H}$ tel que, pour tout point étiqueté $(x^{(train)}, y^{(train)})$, $h(x^{(train)})$ est "aussi proche que possible" de $y^{(train)}$ tout en évitant son surajustement[12]. Plus formellement, dans l'étape d'apprentissage, l'objectif est de trouver le prédicteur optimal $h^* \in \mathscr{H}$, c'est-à-dire un minimiseur du risque théorique Eq. (1.1). Le risque théorique étant incalculable en pratique, nous nous concentrons sur $\hat{h}$ une fonction qui minimise sa forme empirique Eq. (1.2) à la place. Nous appelons cette minimisation la *Minimisation du risque empirique* (ERM) :

$$\hat{h} = \underset{h\in\mathscr{H}}{\operatorname{argmin}}\, \hat{R}_{train}(h)$$

où

$$\hat{R}_{train}(h) := \frac{1}{L}\sum_{i=1}^{L} l\left(h(x_i^{(train)}), y_i^{(train)}\right)$$

Après la phase d'apprentissage, nous nous attendons à ce que la minimisation du risque empirique soit approximativement équivalente à la minimisation du risque réel :

$$\hat{R}_{train}(\hat{h}) \approx R(h^*)$$

[11] c'est-à-dire les données qui n'ont pas encore été traitées

[12] trop de capture des fluctuations et variations aléatoires dans les données d'apprentissage résultant en une mauvaise généralisation de la prédiction d'une donnée non apprise

(iii) **Phase de test.** Cette étape étudie la performance de notre estimateur h sur un ensemble de test $\mathscr{D}^{(test)}$. Notez que cette étape permet de détecter si le modèle sur-apprend ou sous-apprend sur l'ensemble d'apprentissage. De manière plus formelle, cela revient à vérifier que $\hat{h}$ vérifie la condition :

$$\hat{R}_{train}(\hat{h}) \approx \hat{R}_{test}(\hat{h})$$

où

$$\hat{R}_{test}(h) := \frac{1}{T}\sum_{i=1}^{T} l\left(h(x_i^{(test)}), y_i^{(test)}\right).$$

Notez que l'évaluation de l'équité du modèle est également faite dans cette phase. Nous nous référons à la section 1.4.1 pour plus de détails.

Pour un classifieur probabiliste binaire, nous pouvons estimer la probabilité que le modèle h prédise que la classe d'une instance x soit positive : $\mathbb{P}(h(x) = 1|x)$. Un exemple est donné ci-dessous avec la régression logistique.

Exemple de classifieur probabiliste : Régression logistique. Etant donné une instance $x \in \mathscr{X}$, la régression logistique modélise la probabilité conditionnelle de la classe $y \in \{0, 1\}$ par :

$$\mathbb{P}(y|x, w) = \frac{1}{1 + \exp(-yw^T x)}$$

où w est le paramètre du modèle logistique. Nous avons omis ici le paramètre biais pour simplifier les notations. Les paramètres du modèle peuvent être entraînés en maximisant la vraisemblance des données d'apprentissage, c'est-à-dire, minimiser la perte logistique (*logloss*) des instances d'entraînement :

$$\min_w \left\{ \sum_{(x,y)\in\mathscr{D}^{(train)}} \log\left(1 + e^{-yw^T x}\right) + \frac{\lambda}{2} w^T w \right\}$$

avec $\frac{\lambda}{2} w^T w$ un terme de régularisation introduit pour éviter le sur-apprentissage [FLH07]. La régression logistique est un modèle qui peut être entraîné efficacement en utilisant des techniques d'optimisation convexe.

Le sur-apprentissage et le sous-apprentissage sont les causes principales des mauvaises performances des modèles prédictifs générés par les algorithmes de Machine Learning. Ces notions sont étroitement liées aux erreurs d'estimation (*a.k.a.* variance) et d'approximation (*a.k.a.* bias).

Compromis entre le biais et la variance. Étant donné l'ensemble $\mathscr{H}$, désignons par $\hat{h}$ un estimateur (par exemple, l'estimateur ERM), h^* l'estimateur optimal au regard de l'équation (1.2) avec $R^* = R(h^*)$ son erreur. Alors la *excess error* peut être vue comme une décomposition biais/variance :

$$\underbrace{R(\hat{h}) - R(h^*)}_{\text{Excès d'erreur}} = \underbrace{\left\{R(\hat{h}) - \inf_{h\in\mathscr{H}} R(h)\right\}}_{\text{Erreur d'estimation ou variance}} + \underbrace{\left\{\inf_{h\in\mathscr{H}} R(h) - R(h^*)\right\}}_{\text{Erreur d'approximation ou biais}}.$$

Le premier terme est appelé l'erreur d'estimation (ou variance) et le second terme l'erreur d'approximation (ou biais). Ce dilemme biais-variance est un conflit qui existe lorsque nous essayons de minimiser simultanément ces deux sources d'erreur qui empêchent les algorithmes d'apprentissage supervisé de généraliser au-delà de leur ensemble d'apprentissage.

- **Biais d'un estimateur.** Il mesure dans quelle mesure l'estimateur optimal peut être approché dans un espace d'hypothèses $\mathscr{H}$. Le biais est une erreur provenant d'hypothèses erronées dans l'algorithme d'apprentissage. Un biais élevé (un modèle d'apprentissage trop simple) peut conduire à ce que l'algorithme tienne peu compte des relations pertinentes entre les caractéristiques et les sorties cibles (sous-adaptation). En revanche, un biais trop faible peut avoir un impact et conduire à une variance plus élevée.
- **Variance d'un estimateur.** La variance dépend du modèle choisi h. L'erreur d'estimation mesure l'erreur de h par rapport à la borne inférieure des erreurs réalisées par les hypothèses de $\mathscr{H}$. En d'autres termes, la variance est une erreur résultant d'une trop grande sensibilité aux petites fluctuations de la base d'apprentissage. Une variance élevée (modèle d'apprentissage trop complexe) peut conduire à un algorithme qui modélise le bruit aléatoire dans les données d'apprentissage, plutôt que les sorties cibles (overfitting). Une variance trop faible peut conduire à un biais plus élevé.

Ainsi, le modèle est surajusté (resp. sous-ajusté) s'il a une variance (resp. un biais) trop élevée, tandis qu'une valeur trop faible peut conduire à un biais (resp. une variance) plus élevé et donc à une moins bonne performance du modèle. Par conséquent, l'objectif de l'apprentissage automatique supervisé est de choisir un espace d'hypothèses $\mathscr{H}$ qui vérifie un bon compromis entre les erreurs d'approximation et d'estimation.

Comme mentionné ci-dessus, en apprentissage supervisé, pour qu'un modèle soit performant, cela nécessite l'apprentissage d'une quantité considérable de données étiquetées. Cependant, dans la pratique, la plupart des données (non structurées) ne sont pas étiquetées. Nous sommes donc souvent dans le cas où nous disposons d'une petite base d'apprentissage $\mathscr{D}^{(train)}$ suivie d'une (très) grande base de données non étiquetées $\mathscr{D}_{\mathscr{X}}^{(pool)}$. Traditionnellement, tant que les instances de $\mathscr{D}_{\mathscr{X}}^{(pool)}$ ne sont pas étiquetées, nous utilisons exclusivement $\mathscr{D}^{(train)}$ pour calibrer un algorithme d'apprentissage: nous disposons donc de nombreuses instances non étiquetées $\mathscr{D}_{\mathscr{X}}^{(pool)}$ inutilisées alors qu'elles peuvent présenter des informations supplémentaires. Dans cette thèse nous considérons l'utilisation d'ensembles de données étiquetées $\mathscr{D}^{(train)}$ et non étiquetées $\mathscr{D}_{\mathscr{X}}^{(pool)}$ pour calibrer un modèle d'apprentissage **robuste**, **performant** et **éthique**.

1.2.2 Apprentissage semi-supervisé

L'apprentissage semi-supervisé (*semi-supervised learning* en anglais) est à mi-chemin entre l'apprentissage supervisé et non supervisé. Plus précisément, l'apprentissage semi-supervisé peut être vu comme un apprentissage supervisé avec des informations supplémentaires sur la distribution de $\mathscr{X}$. En classification,

son but est le même que l'apprentissage supervisé : prédire la classe d'une donnée via l'entraînement du modèle d'apprentissage sur les données étiquetées tout en capitalisant sur l'abondance des données non étiquetées pour améliorer la tâche d'apprentissage. En d'autres termes, l'apprentissage semi-supervisé est principalement utilisé pour améliorer la performance d'un modèle d'apprentissage lorsque nous n'avons pas assez de données étiquetées mais que les données non étiquetées sont abondantes.

Avec les données non étiquetées de plus en plus abondantes, l'apprentissage semi-supervisé a attiré beaucoup d'attention au cours des dernières décennies et présente un large champ d'applications comme, par exemple, la mise en place des systèmes automatisés de surveillance d'images. En particulier [RLC+21] propose l'application de l'apprentissage semi-supervisé à un système de surveillance automatique des insectes nuisibles. La méthode utilisée comprend un pseudo-étiquetage non supervisé des images d'insectes et l'apprentissage de modèles de classification semi-supervisés pour la reconnaissance des images d'insectes. Nous parlons de pseudo-étiquette pour une étiquette produite par une méthode algorithmique (et non par une intervention humaine). Toutefois, il convient de noter que, bien que l'on s'attende à ce que les performances d'apprentissage soient améliorées par l'exploitation de données non étiquetées, certaines études empiriques montrent qu'il existe des situations où l'utilisation de données non étiquetées peut dégrader les performances [NMTM00a, SNZ08, YP11].

Une notion primordiale en apprentissage semi-supervisé est le *degré de fiabilité* de la prédiction d'une instance non étiquetée $x \in \mathscr{D}_{\mathscr{X}}^{(pool)}$ pour un classifieur $h \in \mathscr{H}$. Dans la suite de ce manuscrit, par abus de langage, nous parlerons de la *fiabilité* de x pour h. Selon la tâche d'apprentissage, il existe de nombreuses mesures pour quantifier cette fiabilité.

Exemple d'une instance fiable (et non fiable) pour un classifieur binaire. Dans un cadre binaire $\mathscr{Y} = \{0, 1\}$ et pour un classifieur probabiliste binaire $h \in \mathscr{H}$,

- l'instance la **plus fiable** dans $\mathscr{D}_{\mathscr{X}}^{(pool)}$ est l'instance qui a sa probabilité (a posteriori) d'être positif la plus éloignée de 0.5 :

$$\widehat{x}_1 = \arg \max_{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}} \{| \mathbb{P}(h(x) = 1|x) - 0.5 |\} \tag{1.3}$$

- Par contraste, l'instance la **moins fiable** dans $\mathscr{D}_{\mathscr{X}}^{(pool)}$ est l'instance qui a sa probabilité (a posteriori) d'être positif la plus proche de 0.5 :

$$\widehat{x}_2 = \arg \min_{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}} \{| \mathbb{P}(h(x) = 1|x) - 0.5 |\}$$

Traditionnellement, l'apprentissage semi-supervisé produit itérativement des (pseudo-)étiquettes[13] (*pseudo-labels* en anglais) à des instances de $\mathscr{D}_{\mathscr{X}}^{(pool)}$ et ainsi enrichit la base d'entraînement $\mathscr{D}^{(train)}$. Notons que le classifieur utilise ses propres prédictions comme (pseudo-)étiquettes. Comme aperçu des méthodologies d'apprentissage semi-supervisé, nous présentons ci-dessous quelques méthodes

[13] Nous rappelons que les (pseudo-)étiquettes sont produites par des méthodes algorithmique.

heuristiques : le auto-apprentissage [Yar95] (*self-training* en anglais) et le co-apprentissage [BM98] (*co-training* en anglais).

Auto-apprentissage. Le auto-apprentissage consiste à entraîner un classifieur sur la base d'entraînement et prédire sur toute la base non étiquetée. Les points les plus "fiables" selon notre classifieur avec leur classe prédite sont ajoutés à la base d'entraînement. Le classifieur est alors ré-entraîné sur la nouvelle base d'entraînement et nous répétons ce processus tant qu'une condition d'arrêt n'est pas atteinte (e.g. convergence de la performance). En d'autres termes, le processus est le suivant :

1. *(Entraînement)* entraîner $h \in \mathcal{H}$ sur les données étiquetées $\mathcal{D}^{(train)}$;
2. *(Requêtage)* requêter les instances $\widehat{x_1}, \ldots, \widehat{x_m} \in \mathcal{D}_{\mathcal{X}}^{(pool)}$ les **plus fiables** de h (e.g. selon eq (1.3)) ;
3. *(Mise à jour)* mettre à jour les bases :

$$\mathcal{D}^{(train)} = \mathcal{D}^{(train)} \cup \{ (\widehat{x_i}, \underbrace{h(\widehat{x_i})}_{\substack{\text{pseudo-} \\ \text{étiquette}}}) \}_{i=1}^{m}$$

et

$$\mathcal{D}_{\mathcal{X}}^{(pool)} = \mathcal{D}_{\mathcal{X}}^{(pool)} - \{\widehat{x_i}\}_{i=1}^{m} \ .$$

Co-apprentissage. Le co-apprentissage consiste à entraîner deux classifieurs basés sur deux différentes bases d'entraînement conditionnellement indépendantes. Plus précisément, nous supposons (1) que l'ensemble des caractéristiques peut être divisé en deux sous-ensembles (tout en gardant le même nombre d'instances), (2) que chaque sous-ensemble de caractéristiques est suffisant pour entraîner un bon classifieur et (3) que les deux sous-ensembles de caractéristiques de chaque instance sont conditionnellement indépendants étant donnée la classe. Les deux classifieurs sont entraînés sur tous les exemples étiquetés et les échantillons fiables pour les deux classifieurs sur les données non étiquetées sont ajoutés aux données étiquetées de manière itérative. En d'autres termes, le processus est le suivant :

1. *(Entraînement)* entraîner $h_1 \in \mathcal{H}$ et $h_2 \in \mathcal{H}$ respectivement sur les données étiquetées $\mathcal{D}_1^{(train)}$ et $\mathcal{D}_2^{(train)}$. Remarquons que $\mathcal{D}_1^{(train)}$ et $\mathcal{D}_2^{(train)}$ ont le même nombre d'instances que $\mathcal{D}^{(train)}$ (mais pas les mêmes caractéristiques) ;
2. *(Requêtage)* requêter les instances $\widehat{x_1}, \ldots, \widehat{x_m} \in \mathcal{D}_{\mathcal{X}}^{(pool)}$ les **plus fiables** (selon par exemple eq.(1.3)) à la fois pour h_1 et h_2;
3. *(Mise à jour)* mettre à jour les bases :

$$\mathcal{D}^{(train)} = \mathcal{D}^{(train)} \cup \{ (\widehat{x_i}, \underbrace{h(\widehat{x_i})}_{\substack{\text{pseudo-} \\ \text{étiquette}}}) \}_{i=1}^{m}$$

et

$$\mathscr{D}_{\mathscr{X}}^{(pool)} = \mathscr{D}_{\mathscr{X}}^{(pool)} - \{\widehat{x}_i\}_{i=1}^{m}$$

Autres méthodologies d'apprentissage semi-supervisé. Dans la littérature il existe une grande variété de méthodes d'apprentissage semi-supervisé, dont notamment :

- des *modèles génératifs* [MU96, NMTM00b] où nous supposons un modèle probabiliste de la forme $p(x,y) = p(y)p(x|y)$ où $p(x|y)$ est un mélange de distributions (e.g. mélange de gaussiennes). Les composants du mélange (i.e. les distributions individuelles) sont calibrés grâce aux données non-étiquetées massives $\mathscr{D}_{\mathscr{X}}^{(pool)}$ et la distribution du mélange (i.e. les probabilités associées à chaque composant) est calibrée grâce aux données étiquetées $\mathscr{D}^{(train)}$.

- des *méthodes non-agnostiques* aux modèles d'apprentissage où, avec les données non étiquetées, l'apprentissage semi-supervisé est utilisé comme une extension de certains algorithmes standard d'apprentissage supervisé comme les SVM (*semi-supervised SVM* en anglais) [BD98, XJZ+09] ou les réseaux de neurones (*deep semi-supervised learning* en anglais) [RMC15, EADvdH17].

- des *méthodes basées sur les graphes*[14] [BC01, ZGL03, BNS06] où les noeuds correspondent à des instances étiquetées et non étiquetées dans l'ensemble de données, et les arêtes reflètent la similarité des instances.

Pour les lecteurs intéressés, une revue de littérature des principales méthodes d'apprentissage semi-supervisé est présentée par [Zhu05], [OHT20] et [VEH20]. Notons que suivant [CSZ09] et [ZG09], l'amélioration itérative d'un modèle d'apprentissage nécessite certains prérequis notamment le fait que les données doivent respecter certaines hypothèses, à savoir que la distribution de $\mathscr{X}$ (i.e. la connaissance de $p(x)$) porte des informations utiles pour l'inférence de $p(y|x)$ $\forall (x,y) \in \mathscr{X} \times \mathscr{Y}$. Si cette dernière hypothèse n'est pas vérifiée alors l'apprentissage semi-supervisé risque de dégrader itérativement la performance en produisant une inférence de plus en plus biaisée.

1.2.3 Vers une boucle de rétroaction entre l'homme et la machine

Bien que, sous certaines conditions, l'approche semi-supervisée peut présenter des avantages significatifs, la production de données pseudo-étiquetées peut présenter quelques inconvénients en pratique:
(1) les étiquettes ne sont pas fiables car elles sont pseudo-étiquetées (ne pas confondre fiabilité des prédictions et fiabilité des étiquettes) ;
(2) les nouvelles instances pseudo-étiquetées ne sont pas suffisamment informatives car nous venons choisir les instances où le modèle d'apprentissage sait déjà comment étiqueter
(3) nous risquons de reproduire les biais d'étiquetage existants, notamment si les étiquettes sont *historiquement* ou *socialement* biaisées, ce qui peut générer des soucis éthiques.

[14] Remarque : ce sont des méthodes non paramétriques

Nous faisons référence à la fiabilité des prédictions comme étant l'incertitude dans les prédictions des modèles d'apprentissage, alors que la fiabilité des étiquettes est basée sur la confiance dans l'acquisition des étiquettes.

Fiabilité des étiquettes. Dans certains domaines comme la détection des cyber-attaques ou la détection de fraudes, l'acquisition des données étiquetées fiables est d'autant plus cruciale en raison des enjeux financiers de telles attaques/fraudes. Nous définissons

- la *détection des cyber-attaques* [CK15, WWZJ19] comme la détection des attaques lancées par des cybercriminels qui tentent d'accéder aux données, aux fonctions ou à d'autres zones restreintes du système d'information sans autorisation. Ces cyber-attaques sont à l'origine des vols d'informations, ou de blocage des systèmes d'informations (rançongiciels).
- la *détection de fraudes* [RP11, DMF18] comme la détection des transactions frauduleuses par carte bancaire (ou par chèque) dans les banques ou la détection des demandes de remboursement frauduleuses dans les assurances. Ces fraudes sont à l'origine de pertes financières importantes pour les entreprises.
- la *non-conformité* des documents [HR18, THN+18] textuels comme la détection de la conformité juridique RGPD[15] des documents textuels. Dans cet exemple l'intervention d'un juriste est indispensable compte-tenu des connaissances juridiques nécessaires pour décider de la conformité de ces documents textuels. Un faux négatif (i.e. un document non conforme detecté conforme) peut engendrer une sanction de la CNIL qui peut aller jusqu'à 4% du chiffre d'affaires des entreprises. À titre d'exemple, des sources de données textuelles volumineuses et potentiellement à risque sont les zones de textes libres utilisées par les télé-conseillers.

Compte tenu des risques trop importants que les étiquettes fausses peuvent engendrer (en particulier en cas de faux-négatifs) les entreprises ont tendance à requérir des interventions humaines pour détecter ces anomalies ou pour valider manuellement ces anomalies détectées par des systèmes d'apprentissage.

Qualité des données étiquetées. Les principales méthodes d'apprentissage semi-supervisé étiquettent les instances qui se trouvent dans la région de certitude de notre classifieur : ces données étiquetées sont donc susceptibles d'être fausses, redondantes ou non-informatives. En effet, l'apprentissage semi-supervisé traditionnel privilégie la quantité (étiqueter le plus possible et espérer avoir un modèle plus performant) des données étiquetées à la qualité (étiqueter les bonnes instances pour avoir un modèle plus performant) de ces dernières. Pour palier ce problème, une approche plus efficace serait d'étiqueter les instances les **moins fiables**, i.e. les plus difficiles à classifier, pour notre modèle. Ces instances, qui se trouvent dans la région d'incertitude de notre modèle, ont l'avantage d'être très informatives pour notre modèle si nous connaissons leur classe. En apprentissage semi-supervisé, cette stratégie n'est

[15] **GDPR** : Règlement général sur la protection des données, un règlement européen qui constitue le texte de référence en matière de protection des données à caractère personnel.

pas concevable car le modèle a un risque beaucoup plus important de générer des étiquettes fausses pour ces instances incertaines. La génération d'étiquettes fausses a pour conséquence des bases de données défectueuses créant ainsi l'effet GIGO[16]. Ainsi, si notre objectif est de sélectionner et étiqueter de manière sûre les instances les plus informatives pour notre modèle, une intervention humaine est nécessaire.

Reproduction des biais dans les étiquettes. Les systèmes d'IA apprennent à prendre des décisions sur la base de données étiquetées. Ces étiquettes peuvent inclure des décisions humaines biaisées ou refléter historiquement des inégalités sociales. La calibration d'un modèle d'apprentissage sur des données étiquetées injustes peut engendrer des prédictions injustes. Reprenons les exemples de la section 1.1.3, pour lesquels l'apprentissage semi-supervisé peut produire (ou répliquer) des biais dans les étiquettes:

- *Biais historique.* L'algorithme de recrutement d'Amazon alimenté par un modèle d'apprentissage favorise implicitement les candidats masculins. La base de données utilisée pour entraîner le système IA vient des modèles de Curriculum Vitae soumis à l'entreprise sur une période de 10 ans dans le secteur de la technologie, et où les hommes étaient très largement sur-représentés.
- *Biais social.* Le logiciel COMPAS utilisé dans les tribunaux pour évaluer la probabilité qu'un défendeur récidive : une personne de couleur se verra systématiquement attribuer un risque plus élevé qu'une personne blanche.

Nous renvoyons à [MMS+21] pour une discussion générale sur les différents biais dans l'apprentissage automatique. Ainsi, un processus d'apprentissage semi-supervisé qui génère des pseudo-étiquettes aux instances va continuer à reproduire l'injustice dans les bases de données (indépendamment de l'évolution de la société et de sa mentalité).

Par conséquent, tous ces inconvénients nous amènent à intégrer un facteur très important dans les processus d'apprentissage semi-supervisé : l'intervention d'un (ou plusieurs) **oracle** (e.g. expert humain) dans l'étiquetage des données. Dans notre étude les données non étiquetées seront utilisées soit pour quantifier $p(x)$, soit pour enrichir la base d'entraînement via un étiquetage actif par un oracle : nous appelons cette dernière procédure apprentissage actif (ou *active learning* en anglais, alias *AL*).

1.3 Apprentissage actif : état de l'art et contributions

Présentons l'état de l'art de l'apprentissage actif ainsi que les contributions de cette thèse dans ce domaine. L'objectif en apprentissage actif est de requêter itérativement l'étiquette de l'instance qui apporte le plus d'information pour notre modèle d'apprentissage. En d'autres termes, le but de l'apprentissage actif est de rendre notre modèle d'apprentissage plus performant avec le moins de données étiquetées possible. Par contraste nous appelons apprentissage passif (*passive learning* en

[16] GIGO : Garbage-In-Garbage-Out.

anglais, alias *PL*) le fait d'étiqueter aléatoirement les instances pour enrichir la base d'entraînement. Présentons un aperçu de la littérature d'apprentissage actif avant de nous concentrer sur notre étude.

1.3.1 Scénarios d'échantillonnage

Il existe plusieurs manières d'accéder aux données non étiquetées, parmi elles nous avons par exemple (1) le mode *hors-ligne* où les données non étiquetées sont accessibles directement en grande quantité: par exemple les avis client sur un produit laissés par les internautes sur un site internet
(2) le mode *en ligne* où les données sont recueillies une à une comme par exemple les mails reçu dans une boîte électronique.
Les deux scénarios dans lesquels l'apprenant actif demande des requêtes se nomment respectivement échantillonnage en ligne[17] (*online sampling* ou *stream-based sampling* en anglais) et échantillonnage hors-ligne (*offline sampling* ou *pool-based sampling* en anglais).

Echantillonnage en ligne. Nous supposons que les données non étiquetées arrivent de manière successive. Ici, le but en AL est de créer un apprenant actif qui décide le requêtage de chaque nouvelle instance $x^{(stream)} \in \mathscr{X}$ reçue. Pour chaque requête acceptée, nous obtenons l'étiquette $y^{(stream)}$ de $x^{(stream)}$ (via l'oracle) et nous ré-entraînons notre modèle d'apprentissage sur la nouvelle base incluant le couple $(x^{(stream)}, y^{(stream)})$ (apprentissage par batch) ou nous réajustons notre modèle précédent en l'entraînant seulement sur le couple $(x^{(stream)}, y^{(stream)})$ (apprentissage en ligne, voir [HSLZ21] pour une revue de la littérature sur ce sujet).

Echantillonnage hors-ligne. Nous supposons que la totalité des données non étiquetées $\mathscr{D}_{\mathscr{X}}^{(pool)} \subset \mathscr{X}$ est disponible. Dans ce scénario, le but de l'apprenant actif serait tout d'abord de classer les instances selon leur informativité (généralement mesuré selon un score que nous appelons *score d'informativité*), puis de requêter l'étiquette $y^{(pool)}$ de l'instance la plus informative $x^{(pool)}$. Enfin nous ré-entraînons notre modèle d'apprentissage sur $(x^{(pool)}, y^{(pool)})$ soit via un apprentissage par batch, soit via un apprentissage en ligne.

Notons que les stratégies d'échantillonnage hors-ligne peuvent être généralisées facilement pour le scénario en ligne. En effet, l'échantillonnage hors ligne évalue et trie l'ensemble des données avant de choisir la meilleure instance tandis que l'échantillonnage en ligne parcourt et évalue les données séquentiellement et requête individuellement les instances selon un seuil de décision de requêtage. Ainsi, au lieu de choisir la meilleure instance parmi un ensemble de données non étiquetées, l'échantillonnage en ligne va venir échantillonner les instances qui ont leur score d'informativité supérieur à un certain seuil (qui est souvent déterministe). Le but pour les deux scénarios d'échantillonnage est de retenir l'ensemble de requêtes qui maximise la performance de notre classifieur. Nous présenterons les principales méthodes d'AL les plus utilisées dans la section 1.3.3.

[17] A ne pas confondre avec l'apprentissage en ligne qui consiste à mettre à jour itérativement le modèle exclusivement via le flux entrant. Le flux peut être une donnée ou un paquet de données.

Dans la littérature, chaque stratégie d'échantillonnage dépend souvent du problème de modélisation que nous étudions : les techniques d'échantillonnage pour un problème de classification seront souvent différentes de celles pour une régression en raison de la nature des réponses des modèles. Contrairement à la régression, un modèle probabiliste de classification donne souvent une réponse directement interprétable (par exemple une probabilité a posteriori qu'une classe donnée soit correcte), ce qui conduit à des choix heuristiques naturels pour la mesure d'informativité.

En pratique, l'AL voit son utilité dans l'étiquetage des données non structurées (des données non formatées comme les images, textes, sons, ...) car ces dernières souffrent plus souvent d'un manque d'étiquetage en raison de sa forte complexité en temps et en coût. En effet, un étiquetage d'une donnée non structurée exige l'intervention d'un expert humain rigoureux (d'où le *coût*) qui doit analyser et étiqueter les données une à une (d'où le *temps*), notamment lorsque la donnée qui nécessite d'être analysée est sous format son, vidéo ou texte. Ainsi, dans cette thèse l'AL sera étudié sur des données non structurées, ce qui nous contraint à nous concentrer sur des problématiques de classification, plus précisément la **classification binaire et multi-classes**. Enfin, nous supposons que les données non étiquetées sont abondantes mais l'étiquetage de ces dernières est difficile, coûteux et long : nous étudions donc le **scénario hors-ligne**.

1.3.2 Apprentissage actif hors-ligne

Nous rappelons que le but en AL est de construire une base d'entraînement $\mathscr{D}_{active}^{(train)}$ plus informative pour entraîner notre algorithme d'apprentissage qu'une base d'entraînement $\mathscr{D}_{passive}^{(train)}$ construite via un requêtage aléatoire (apprentissage passif). Autrement dit, si nous notons $\hat{h}_{active}, \hat{h}_{passive} \in \mathscr{H}$ tels que $\hat{h}_{active}$ soit entraîné sur $\mathscr{D}_{active}^{(train)}$ et $\hat{h}_{passive}$ entraîné sur $\mathscr{D}_{passive}^{(train)}$ avec $\left|\mathscr{D}_{active}^{(train)}\right| = \left|\mathscr{D}_{passive}^{(train)}\right|$ alors

$$\hat{R}(\hat{h}_{active}) \leq \hat{R}(\hat{h}_{passive}).$$

Sans perte de généralité, dans toute la suite de cette thèse, nous considérons seulement le cas où l'échantillonnage se fait en hors-ligne. Dans ce contexte, l'échantillonnage d'une instance à étiqueter dépend donc d'un score d'informativité, que nous notons I. Des exemples de fonction I seront donnés en section 1.3.3. Le processus d'AL est donné par l'algorithme 1.
Dans les prochaines sections nous introduirons les différentes approches les plus "classiques" en AL pour définir I.

1.3.3 Méthodologies

Commençons par une méthode plutôt naïve et simple qui est néanmoins l'une des techniques les plus utilisées dans la littérature : le critère d'échantillonnage basé sur l'incertitude d'un modèle d'apprentissage.

Échantillonnage basé sur l'incertitude. Cette approche considère que les instances les plus informatives sont celles situées dans la région d'incertitude du modèle d'apprentissage. Par conséquent,

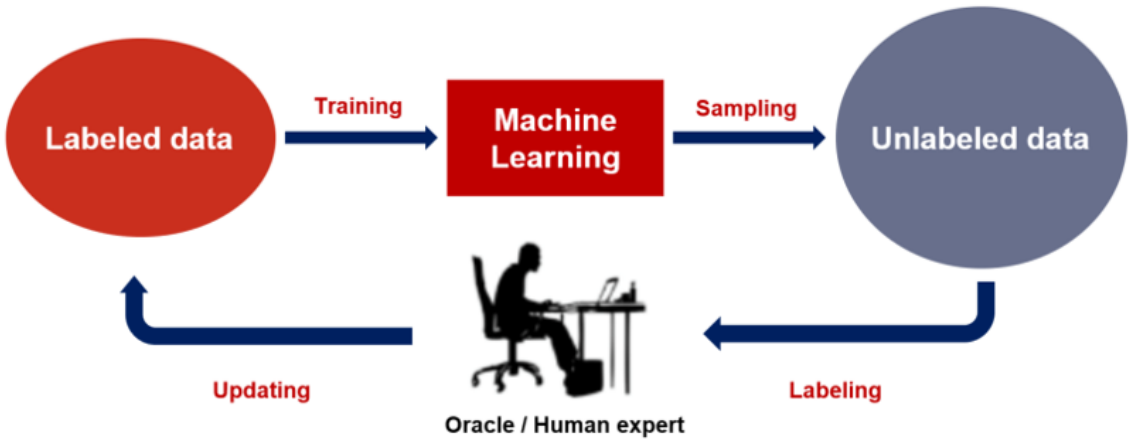


Figure 1.2: Schéma d'apprentissage actif hors-ligne

Algorithm 1 Apprentissage actif hors-ligne

Input: Soient $\mathscr{D}_{\mathscr{X}}^{(pool)}$ l'ensemble des données non étiquetées, $\mathscr{D}^{(train)}$ l'ensemble des données étiquetées et $h \in \mathscr{H}$ un prédicteur. Alors un processus d'AL est :

Etape 1 : (Entraînement) Entraîner h sur la base d'entraînement $\mathscr{D}^{(train)}$

Etape 2 : (Requêtage) l'apprenant actif requête l'instance $\hat{x} \in \mathscr{D}_{\mathscr{X}}^{(pool)}$ qui maximise un **score d'informativité** $I(x,h)$ i.e.

$$\hat{x} = \arg \max_{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}} \{I(x,h)\}$$

Etape 3 : (Mise-à-jour) Nous mettons ensuite à jour la base d'entraînement et l'ensemble des données non étiquetées : si nous notons $y^{(oracle)}$ la classe de $\hat{x}$, alors

$$\mathscr{D}^{(train)} = \mathscr{D}^{(train)} \cup \{(\hat{x}, y^{(oracle)})\}$$

et

$$\mathscr{D}_{\mathscr{X}}^{(pool)} = \mathscr{D}_{\mathscr{X}}^{(pool)} - \{\hat{x}\}$$

Etape 4. Tant que nous n'atteignons pas un critère d'arrêt (par exemple épuisement du budget d'étiquetage ou convergence de la performance), nous réitérons ce processus (revenir à l'étape 1.).

Output: L'estimateur final h.

l'idée est de requêter puis d'apprendre l'échantillon le plus incertain selon l'estimateur. Cette approche tend à éviter le requêtage d'instances redondantes puisque le modèle appris sur un point de données incertain sera probablement plus certain. Formellement, étant donné le prédicteur actuel $h_t \in \mathscr{H}$ (t étant la $t-ième$ mise à jour du modèle) et une instance $x \in \mathscr{X}$, nous désignons par $p_{t,x}(y) := \mathbb{P}(h_t(x) = y|x)$ la réponse probabiliste de classer x comme y selon le modèle h_t.

- en **classification binaire**, une méthode naturelle consiste à échantillonner les instances les moins fiables de h (définition présentée dans la section 1.2.2),

$$\hat{x} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmin}} \left\{ \left| p_{t,x}(y) - \frac{1}{2} \right| \right\}$$

- en **classification multi-classes**, soit $K > 2$ le nombre de classes, et donc $\mathscr{Y}$ est l'ensemble $[K] = \{1, 2, \ldots, K\}$, l'entropie de Shannon (définie dans [Sha48]) attribue à une instance x un score d'incertitude selon l'uniformité de la distribution de probabilité postérieure des étiquettes $p_{t,x} = \left(p_{t,x}(1), p_{t,x}(2), \cdots, p_{t,x}(K)\right)$: une distribution proche de la loi uniforme souligne une prédiction incertaine tandis qu'une distribution éloignée résulte d'une prédiction plus fiable. Pour toute instance x et un temps fixe t, on note le score d'incertitude $H_{t,x} := H_{t,x}(Y)$ avec Y la variable aléatoire discrète sur $\mathscr{Y}$ où chaque événement $\{Y = y\}$ se produit avec une probabilité $p_{t,x}(y)$. Le score d'incertitude basé sur l'entropie est défini par $H_{t,x} = -\sum_{k=1}^{K} p_{t,x}(k) \log p_{t,x}(k)$ et nous requêtons l'instance qui vérifie

$$\hat{x} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ H_{t,x} \right\} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ -\sum_{k=1}^{K} p_{t,x}(k) \log p_{t,x}(k) \right\}.$$

Échantillonnage basé sur le désaccord. Au lieu de se fier à la mesure d'incertitude basée sur un seul modèle, l'échantillonnage basé sur les désaccords propose une méthode plus robuste en combinant le résultat de plusieurs modèles d'apprentissage tous (légèrement) différents les uns des autres (méthodes dites *ensembling*). L'idée décrite dans [SOS92] est de s'appuyer sur un ensemble de modèles pour "voter" l'informativité de chaque instance. Cet ensemble de modèles est appelé *Comité*. Pour un comité, une instance informative est caractérisée par le plus grand désaccord de vote entre les modèles. Pour une instance donnée, le vote d'un modèle peut être caractérisé soit par sa prédiction d'étiquette (alias *vote fort*), soit par sa probabilité postérieure d'étiquette (alias *vote faible*). Cette approche est appelée *Query-By-Committee* (alias *QBC*). QBC nécessite deux composants principaux, à savoir la construction du **comité** et la définition de la **mesure de désaccord**. [AM98] propose de choisir un modèle d'apprentissage (SVM par exemple) et d'utiliser des méthodes d'ensemble telles que Bagging ou Boosting pour construire le comité. Il existe différentes mesures de désaccord pour évaluer la dispersion des votes pour les problèmes de classification multi-classes. Par exemple [DE95] propose le *vote par entropie*, une méthode basée sur l'entropie combinée avec un vote fort de comité :

$$\hat{x} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ -\sum_{k=1}^{K} \frac{v_{t,x}^{\text{committee}}(k)}{C} \log \frac{v_{t,x}^{\text{committee}}(k)}{C} \right\}$$

où $v_{t,x}^{\text{committee}}(k) := \sum_{c=1}^{C} \mathbb{1}_{\{h_t^c(x)=k\}}$ est le nombre de votes forts du comité pour l'étiquette k étant donné l'instance x.

Échantillonnage basé sur le changement de modèle. L'idée de ces approches est de choisir l'instance qui donne le plus de changement (ou d'impact) sur notre modèle d'apprentissage si nous connaissons son étiquette. Dans la littérature actuelle, lorsque nous connaissons son étiquette, une instance candidate peut avoir un impact sur le modèle principalement de deux manières :

- **Impact sur les paramètres du modèle.** L'idée est de requêter les instances qui peuvent modifier le modèle d'apprentissage $h_t \in \mathscr{H}$ autant que possible. Cela peut être fait en évaluant le changement des paramètres du modèle entre le modèle mis à jour h_{t+1} et le modèle actuel h_t. Intuitivement, si une instance est capable de modifier considérablement les paramètres d'un modèle, alors cette instance contient des informations sur la distribution sous-jacente $\mathscr{X}$ qui ne se trouvent pas (ou rarement) dans la base d'apprentissage. [SC08b] propose une stratégie qui s'applique à tous les modèles d'apprentissage nécessitant le calcul du gradient d'une fonction de perte pendant l'apprentissage (par exemple, l'apprentissage par descente de gradient). Considérons l_t une fonction de perte par rapport au modèle $h_t \in \mathscr{H}$ et ∇l_t son gradient. Le degré de changement du modèle peut être mesuré par $|||\nabla l_t(\mathscr{D}^{(train)})|||$ la norme euclidienne du gradient de la fonction de perte évaluée sur les données d'entraînement.

- **Impact sur la performance du modèle**. L'idée est de requêter les instances qui peuvent réduire l'erreur de généralisation. Nous pouvons réduire l'erreur de prévision en estimant directement cette erreur de manière empirique. Par exemple [RM01] propose d'échantillonner l'instance qui minimise l'espérance de l'erreur de généralisation puisque la classe de l'instance est actuellement inconnue.

Échantillonnage basé sur la représentativité. Certaines des stratégies ci-dessus peuvent proposer des instances non représentatives de la distribution de $\mathscr{X}$, ce qui peut entraîner une baisse des performances du modèle. Par exemple, il est possible que dans notre ensemble de pools $\mathscr{D}_{\mathscr{X}}^{(pool)}$, nous ayons des anomalies telles que des *outliers* qui ne sont pas représentatives de la distribution de $\mathscr{X}$ mais qui peuvent être considérées comme informatives au sens des approches d'échantillonnage présentées ci-dessus[18]. Une composante d'informativité seule n'est pas suffisante et certaines stratégies proposent d'ajouter une composante de représentativité (une instance doit ainsi vérifier un bon compromis entre informativité et représentativité). Un exemple de cette méthodologie, appelée *Densité d'information*, est présenté par [SC08a] qui calcule un score de représentativité en mesurant la similarité moyenne entre une instance et les instances de $\mathscr{D}_{\mathscr{X}}^{(pool)}$. [HJZ14] propose une stratégie nommée QUIRE (*Query Informative and Representative Examples*) qui mesure (et combine) l'informativité et la représentativité d'une instance par l'incertitude de prédiction d'un modèle d'appprentissage : l'informativité d'une instance x est mesurée par l'incertitude de prédiction du modèle entraîné sur $\mathscr{D}^{(train)}$, tandis que la représentativité de x est mesurée par l'incertitude de prédiction du modèle basé sur $\mathscr{D}_{\mathscr{X}}^{(pool)}$ (avec les pseudo-étiquettes prédites par apprentissage semi-supervisé). [WZS19] propose un processus AL qui,

[18] Un outlier peut se trouver dans la zone d'incertitude du modèle ou peut avoir un impact significatif sur les paramètres du modèle après avoir été étiqueté.

à chaque itération d'AL, requête les instances S qui combine l'informativité et la représentativité en un seul problème d'optimisation dont la fonction objectif est :

$$\min_{S \subset \mathscr{D}_{\mathscr{X}}^{(pool)}, |S|=b} R(S) + \lambda LC(S) \tag{1.4}$$

où b est la taille du lot fixée par l'utilisateur, R est le score de représentativité, LC le score de certitude qui assure une limite inférieure pour les instances à faible certitude (nommé Lower-Bounded Certainty ou *LBC*) et λ le paramètre de compromis entre le score R et LC.

Échantillonnage basé sur l'architecture des réseaux neuronaux. Les méthodes suivantes sont conçues pour les architectures de réseaux de neurones et peuvent être divisées en deux catégories : les approches d'échantillonnage basées sur l'incertitude et les approches d'échantillonnage basées sur la représentativité.

- **Incertitude des instances.** Des études récentes ont montré que les incertitudes peuvent être estimées par l'introduction de méthodes bayésiennes dans les réseaux de neurones. Nous les appelons les méthodes *Bayesian Deep Learning*. L'application de méthodes bayésiennes sur les réseaux neuronaux afin d'approcher le plus possible la vraie distribution de la probabilité postérieure des paramètres $p(w|\mathscr{D}^{(train)})$ (avec w les paramètres des réseaux neuronaux) a été largement étudiée dans la littérature. Récemment, [GG15b] propose de s'appuyer sur les avancées récentes de *Bayesian Deep Learning* pour l'estimation de l'incertitude du modèle d'apprentissage.

- **Représentativité des instances.** Les instances sélectionnées doivent représenter équitablement la distribution sous-jacente $\mathbb{P}_{\mathscr{X}}$. Basé sur un réseau neuronal profond, [YQC+17] propose un algorithme qui, durant la phase d'apprentissage, projette les instances dans un autre espace, où la similarité peut être mesurée plus précisément. Ainsi, dans la phase de requêtage, l'apprenant actif sélectionne des instances representatives (et incertaines).

Echantillonnage basé sur l'apprentissage par renforcement. En AL, une méthode basée sur l'apprentissage par renforcement (*reinforcement learning* en anglais, alias *RL*) remplace les critères de stratégie de requête élaborés heuristiquement par des critères appris par la machine. Dans le scénario hors-ligne, les auteurs de [LBH18] considèrent un processus de décision de Markov dans lequel l'AL correspond à la décision de sélection des instances les plus informatives de $\mathscr{D}_{\mathscr{X}}^{(pool)}$: la stratégie d'AL est d'abord apprise sur des simulations (par exemple des données où l'étiquette d'une partie des données est cachée puis révélée pour disposer d'un oracle automatique) et elle est ensuite appliquée à des scénarios AL réels. Notons que plus les tâches du scénario réel sont liées à celles utilisées pour former la stratégie AL, plus celle-ci sera efficace. D'autres recherches, comme [BST17] ou [PDWH18] utilisent la (deep-) RL non seulement pour apprendre les stratégies d'étiquetage, mais aussi pour calibrer l'agent de sorte à généraliser à travers différents ensembles de données.

1.3.4 Apprentissage actif en mode batch

Traditionnellement, chaque étape d'AL est centrée sur la sélection d'une instance à étiqueter avant de réajuster le modèle d'apprentissage à l'ensemble des données étiquetées car l'étiquetage d'un trop grand nombre d'instances en une seule étape AL peut réduire la qualité de la boucle de rétroaction de l'interaction homme-machine. Notons de plus que le requêtage de données basée sur l'incertitude peut souffrir du problème de chevauchement, c'est-à-dire que les k premiers échantillons classés par l'incertitude peuvent être similaires. Cependant, en pratique, ce cadre soulève des problèmes de vitesse et d'adaptabilité. En effet, une des préoccupations de l'AL est le temps d'attente des experts humains pour les prochaines instances à étiqueter : un processus plus adaptable serait de laisser les annotateurs étiqueter plusieurs instances à la suite. Ce problème devient apparent lorsque (1) de nombreux oracles sont disponibles pour une annotation parallèle ou lorsque (2) le modèle d'apprentissage est complexe (par exemple les modèles d'apprentissage profond qui nécessitent parfois des heures ou des jours de réentraînement).

Nous appelons ce problème le coût du délai de réentraînement. Un processus plus approprié consiste donc à requêter l'étiquette d'un lot (ou *batch*) d'instances à chaque étape de l'AL. Ce cas particulier d'AL est appelé AL en mode batch (*Batch Mode Active Learning* en anglais alias *BMAL*). Dans ce contexte, une méthode BMAL bien définie permet d'établir un compromis entre la précision et le coût du délai de réapprentissage. Pour une étape AL donnée, ce compromis est déterminé par la taille du batch de requêtes. Dans la littérature, il existe trois approches principales du BMAL :

1. *les méthodes de classement* qui classent les instances non étiquetées, généralement en fonction de leur caractère informatif ou représentatif, et sélectionnent les meilleures d'entre elles. Cette méthodologie peut être utilisée avec les stratégies ci-dessus qui affectent un score d'importance par instance [DE95, SC08b, RM01, SC08a, GG15b].

2. *les méthodes par objectifs* [GS07, CBP14, CBS+15, YMN+15, WZS19] formulent et résolvent un problème d'optimisation aboutissant à un lot d'instances les plus informatives (ou représentatives) comme par exemple l'approche par batch basée sur le *LBC* de [WZS19] présentée dans la section précédente.

3. *les méthodes par cluster* [PB12, SS18] réduisent l'espace non étiqueté à un sous-ensemble contenant des instances informatives (généralement incertaines en fonction des réponses du modèle) et sélectionnent certaines instances représentatives comme par exemple l'approche *core-set* de [SS18] qui propose des méthodes basées sur la densité. Les auteurs proposent de définir le problème d'AL comme une sélection *core-set* : trouver un petit sous-ensemble d'un grand ensemble de données étiquetées tel qu'un modèle appris sur le petit ensemble de données est compétitif sur l'ensemble des données étiquetées. L'idée est de choisir c centroïdes de telle sorte que la plus grande distance entre ces centroïdes et le reste des données non étiquetées soit aussi petite que possible. Cependant, les approches *core-set* nécessitent de calculer une grande matrice de distance sur les données non étiquetées, ce qui entraîne une complexité de calcul élevée.

1.3.5 Apprentissage actif pour l'Assurance

En pratique, malgré les progrès récents de l'apprentissage automatique, les actuaires préfèrent souvent utiliser les "modèles actuariels" classiques tels que le modèle linéaire généralisé (a.k.a. *Generalized Linear Models* ou GLM[19]) comme par exemple pour l'optimisation de la tarification des produits d'assurance [FDM14, SDP18]. Ce n'est que depuis quelques années que les actuaires ont commencé à étudier de nouvelles approches de Machine Learning pour les tâches liées à l'assurance et l'apprentissage actif n'est que marginalement étudié par la communauté actuarielle, hormis quelques travaux comme [KGM10, RZS19]. [KGM10] propose un processus d'apprentissage actif pour prédire et prévenir les erreurs dans le traitement des demandes d'indemnisation de l'assurance maladie. D'après les auteurs, les erreurs de paiement commises par les compagnies d'assurance lors du traitement des demandes peuvent augmenter les coûts de l'assurance maladie. Ces erreurs entraînent souvent un effort administratif supplémentaire pour retraiter la demande et l'apprentissage actif tente de réduire cet effort. [RZS19] propose un cadre AL pour lutter contre la fraude sur les pensions de retraite. En Chine, si un retraité est décédé mais que sa famille n'en a pas informé l'institution d'assurance sociale, cette dernière continuera à verser des pensions au retraité. Cette détection de fraude sur les pensions se fait souvent manuellement et l'apprentissage actif pourrait réduire considérablement l'éventail des populations à haut risque.

Les raisons suivantes peuvent expliquer pourquoi l'adoption de l'IA dans les sciences actuarielles a été plus lente que dans d'autres domaines:

- *Problème d'explicabilité des modèles.* Les modèles ML sont des modèles de type boîte noire qui sont excellents pour la prédiction, mais moins bons pour l'interprétation. Lorsqu'il s'agit de modèles de tarification d'assurance, l'interprétation est cruciale pour l'approbation réglementaire. En effet, avec ces algorithmes en boîte noire, il est difficile de s'assurer que le modèle ne discrimine pas et n'induit pas de biais involontaires, ce qui peut exposer l'entreprise à des risques opérationnels et de réputation. C'est pourquoi les GLM, qui sont plus explicables, restent la technique la plus utilisée dans la plupart des modèles de tarification.

- *Actuaires: spécialistes des données structurées.* Traditionnellement, la science actuarielle est spécialisée dans les données structurées, tandis que la science des données est plus apte à travailler avec des données non structurées. Aujourd'hui, le secteur de l'assurance reconnaît l'importance des données non structurées. Par exemple, les compagnies d'assurance ont identifié une opportunité d'utiliser les données télématiques[20] pour ajuster les primes en fonction du comportement du conducteur (voir par exemple le concept *pay as/how you drive* [TYV16]).

Cependant, avec

[19] Exemple de GLM : la régression logistique présentée dans la section 1.2.1.

[20] La télématique est un outil qui permet de géolocaliser les véhicules en utilisant la technologie GPS et les systèmes d'IoT pour tracer leurs trajectoires sur une carte informatisée.

- le développement des nouveaux outils pour améliorer l'interpretabilité [GBY+18, CPC19, ESAMS21] et l'équité [ADW19, CDH+19, CJS+20, DEHH21] des modèles ML;

- et l'identification de nouveaux use-cases avec les données non structurées, tels que la conformité des documents textuels avec la réglementation GDPR, la tarification automobile à l'aide de données télématiques ou la tarification de l'assurance habitation à l'aide de données satellitaires (ces applications nécessitent traditionnellement la collecte de données massives étiquetées)

l'apprentissage actif semble être un atout considérable dans l'adoption des modèles ML en assurance.

1.3.6 Contributions

Présentons à présent une vue d'ensemble des principales contributions du chapitre 2 et du chapitre 3 à l'apprentissage actif.

1.3.6.1 Un aperçu des méthodes d'apprentissage actif pour l'assurance en tenant compte de l'équité algorithmique.

Le chapitre 2 aborde et résout certains défis liés à l'adoption de l'apprentissage automatique dans l'assurance avec la démocratisation du déploiement des modèles :

1. Les données non étiquetées étant généralement très abondantes dans le secteur de l'assurance, le premier défi consiste à réduire l'effort d'étiquetage (donc à se concentrer sur la qualité des données). À cette fin, le chapitre 2 présente diverses méthodologies classiques d'apprentissage actif avant d'étudier leur impact empirique sur des ensembles de données synthétiques et de données actuarielles réelles.

2. Un autre défi majeur dans le domaine de l'assurance est la question de l'équité dans les inférences de modèles (voir la section 1.1.3). Nous introduisons et intégrons la procédure d'équité *post-processing* pour les tâches multi-classes proposées dans un de mes articles [DEHH21] dans le processus d'apprentissage actif. La procédure d'équité *post-processing* proposé dans [DEHH21] fera l'objet du chapitre 4. En d'autres termes, le chapitre 2 aborde la question de l'équité algorithmique comme une extension de la configuration AL.

Des analyses numériques sur des jeux de données injustes mettent en évidence que
(1) la configuration AL (sans contrainte d'équité) semble être plus performante (en termes d'*Accuracy* ou de F_1-*score*) que la configuration PL ;
(2) la configuration AL (sans contrainte d'équité) semble être robuste aux données déséquilibrées ;
(3) la configuration AL avec contrainte d'équité présente un bon compromis entre la précision et l'équité des modèles ML.

Analyse de la précision. Les études numériques (Figure 1.3) montrent que les stratégies d'apprentissage actif (AL) sont plus performantes que l'apprentissage passif (PL) en échantillonnant des données de meilleure qualité pour le modèle d'apprentissage automatique étudié (XGBoost).

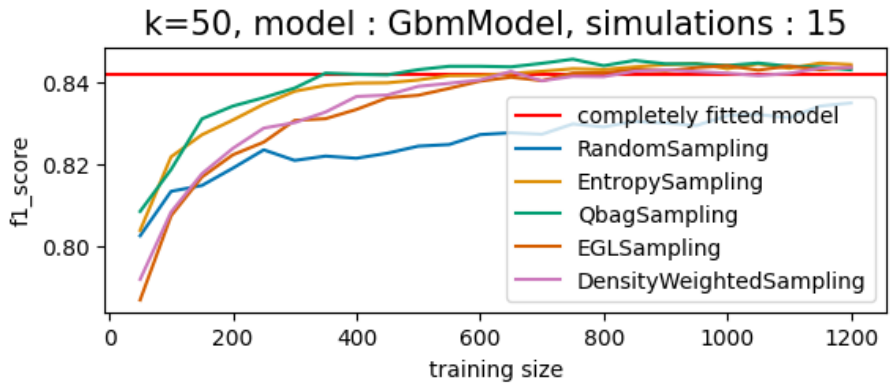
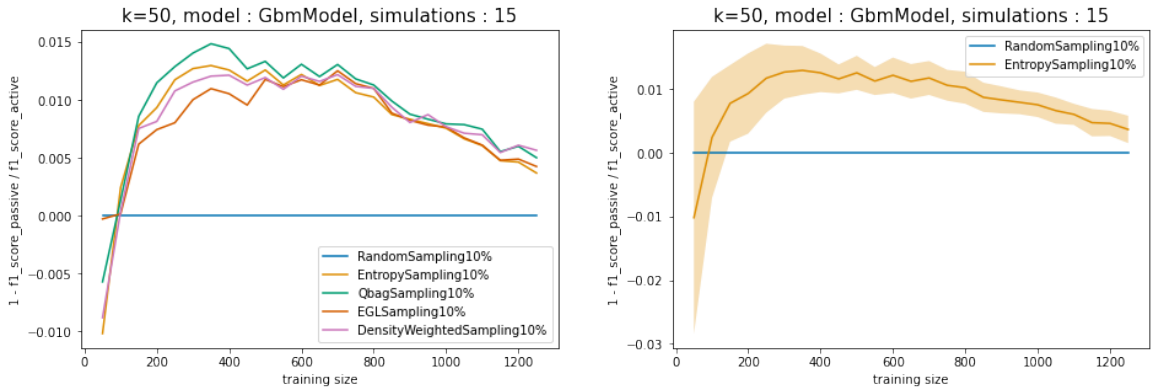


Figure 1.3: Performance de PL et AL sur des données équilibrées
Performance des procédures de classification en termes de **F1-score** pour l'estimateur **XGBoost** par rapport à l'ensemble d'entraînement construit itérativement par les méthodes PL et AL mentionnées ci-dessus. À chaque itération, nous requêtons $k = 50$ instances sur un jeu de données **équilibré** (30%).

Analyse de la robustesse (aux données déséquilibrées). La figure 1.4 met en évidence la performance des procédures AL en évaluant l'écart de performance entre AL et PL:

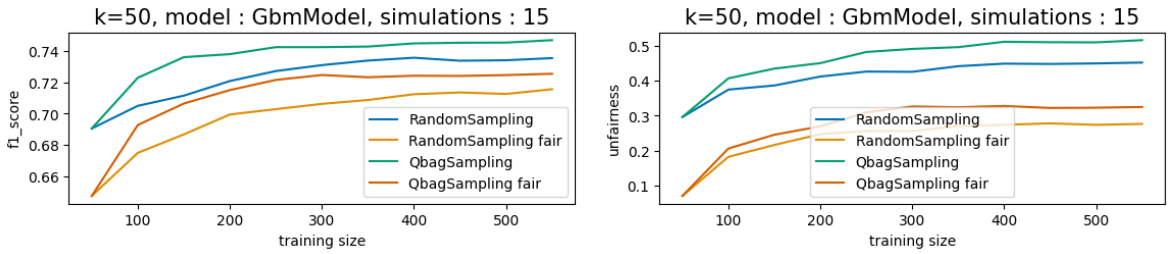
$$\text{GAP} = 1 - \frac{\text{f1\_score\_passive}}{\text{f1\_score\_active}}$$

où f1_score_passive (resp. f1_score_active) est le F_1-score du processus PL (resp. AL). Les écarts de performance entre les méthodologies AL et PL montrés dans la figure 1.4 indique la robustesse des procédures AL sur des données déséquilibrées. Plus précisément, plus les données sont déséquilibrées, plus les performances de AL sont proches de celles de PL. Notons que [EHBG07] montre que AL est performant dans un cas légèrement déséquilibré mais peut être inefficace dans un cas fortement déséquilibré.



(a) (b)

Figure 1.4: Performance des méthodes PL et AL sur un jeu de données déséquilibré (10%). Performance des procédures de classification en termes de **F1-score** pour l'estimateur **XGBoost** par rapport à l'ensemble d'apprentissage construit itérativement par les méthodes PL et AL sur un jeu de données **déséquilibré** (10%) où à chaque itération nous requêtons $k = 50$ instances. Chaque résultat correspond à la moyenne sur 15 simulations et la zone colorée à l'écart-type.



(a) F1-score analysis (b) Unfairness analysis

Figure 1.5: Model performance on UNPS Dataset.
La performance du modèle (précision et injustice) par rapport à la taille des données d'entraînement. Nous étudions à la fois les modèles justes et injustes. Chaque résultat correspond à la moyenne sur 15 simulations où à chaque itération nous requêtons $k = 50$ instances.

Analyse de l'équité. La figure 1.5 illustre la précision (figure 1.5a) et l'équité (figure 1.5b) d'un modèle ML via des procédures AL et PL. Bien que le modèle soit plus précis (en termes de F_1-score) avec l'apprentissage active, la figure 1.5b montre qu'une stratégie passive (*RandomSampling*) construit un modèle plus équitable qu'une stratégie active (*query-by-bagging*). Cependant notons que l'algorithme d'équité post-processing proposé par [DEHH21] réduit l'injustice. L'apprentissage actif avec un algorithme de réduction de l'injustice en post-processing peut être un bon compromis entre précision et équité du modèle. En effet, en termes de précision, la figure 1.5a montre que la stratégie passive sans contrainte d'équité (*RandomSampling*) et la stratégie active avec contrainte

d'équité (*QbagSampling fair*) sont compétitives.

1.3.6.2 Apprentissage actif en mode batch à taille variable.

Contexte. À notre connaissance, en BMAL, peu de travaux se concentrent sur la séquence des tailles optimales de batch (ou lot). Si certains articles mettent en évidence la meilleure taille à définir à l'avance et d'autres optimisent une taille de lot pour une itération AL donnée [CBP14, LGW18], aucun des travaux précédents ne considère la taille du lot comme un paramètre à optimiser en tenant compte du coût d'attente globale de la procédure AL. Le chapitre 3 souligne l'importance de la taille de lot AL en tant qu'impact direct sur la performance du modèle. Plus précisément, nous considérons le choix de la taille du lot (dans un BMAL) comme un problème de contrôle stochastique où la qualité du modèle est un processus stochastique contrôlé par la taille du lot. L'objectif de ce chapitre est de trouver la séquence optimale de taille de batch d'apprentissage actif telle que la procédure maximise la performance du modèle tout en réduisant le nombre d'itérations d'AL. Pour ce faire nous définissons la procédure BMAL comme un cadre de Processus de décision markovien (ou *Markov Decision Process*, MDP en anglais).

Pour résoudre un tel problème de contrôle optimal, une approche basée sur le principe de programmation dynamique (PPD) est utilisée. Rappelons que ce principe, initié par Bellman [Bel58, Bel66], conduit à une équation différentielle partielle (EDP) non linéaire du second ordre, appelée équation de Hamilton Jacobi Bellman (HJB) associée à une fonction valeur.

Formulation du problème. Pour bénéficier de la puissance du calcul stochastique, nous formulons le problème dans un cadre à temps continu. Nous considérons un espace de probabilité filtré $(\Omega, \mathbb{F}, \mathbb{P})$ où la filtration $\mathbb{F} = (\mathscr{F}_t)_{t\geq 0}$ est la filtration naturelle (complète et continue à droite) engendrée par un mouvement brownien unidimensionnel $(W_t)_{t\geq 0}$.

Un cadre MDP est défini par $(\mathscr{S}, \mathscr{A}, \mathscr{T}, \gamma)$ avec $\mathscr{S}$ l'ensemble des états possibles, $\mathscr{A}$ l'ensemble des actions, $\mathscr{T}$ la fonction de transition et γ la fonction de récompense. Au temps t, les processus d'état (Q_t, B_t) sont des processus $\mathbb{F}$-adaptés à valeurs dans $\mathscr{S}$. Q_t correspond à la performance (ou qualité) du modèle et B_t au budget total consommé au temps t. Le budget consommé $B_t = \int_0^{t\wedge\tau} b_s ds$ est un nombre réel positif compris entre 0 (aucune instance requêtée) et B_{MAX} (épuisement du budget d'annotation), où b_t est le taux d'étiquetage au temps t et τ est le $\mathbb{F}$-temps d'arrêt défini par

$$\tau = \inf\{t \geq 0 \quad | \quad B_t = B_{MAX} \text{ or } Q_t = 0 \text{ or } Q_t = 1\}.$$

Le contrôle $(b_t)_t \in \mathscr{A}$ correspond à la taille du batch dans lequel l'apprentissage actif requête au temps t l'étiquette des b_t instances dans le pool-set. Le processus de performance $(Q_t)_t$ est supposé satisfaire une équation différentielle stochastique (EDS) dirigée par le mouvement brownien $(W_t)_t$, dont les coefficients (μ, σ) dépendent à la fois de b_t le taux de données annotées et de B_t le nombre total de données annotées.

La dynamique des processus d'état sont définis par

$$\begin{cases} dQ_t = \mu(B_t, b_t) \cdot Q_t(1-Q_t) \cdot dt + \sigma(B_t, b_t) \cdot Q_t(1-Q_t) \cdot dW_t \\ dB_t = b_t \cdot dt \end{cases} \tag{1.5}$$

Le chapitre 3 montre qu'il est naturel de choisir les fonctions σ et μ croissantes en b.
Comme il est usuel, la récompense γ_t est une fonction d'utilité régulière concave U de Q_t : $\gamma_t = U(Q_t)$. La fonction d'utilité U modélise l'aversion au risque de l'utilisateur concernant la performance du modèle. Par ailleurs, le coût est supposé être une fonction convexe $c : b \to \mathbb{R}$ de la taille du lot b. A partir de ces formalisations, nous formulons notre problème d'optimisation comme suit

$$\sup_{(b_s)\in\mathscr{A}} \mathbb{E}\left[U(Q_\tau) - \int_0^\tau c(b_s)ds\right]. \tag{1.6}$$

Nous résolvons l'équation de HJB correspondante, ce qui nous conduit à un BMAL dynamique amélioré.

Principe de programmation dynamique et équation de Hamilton Jacobi Bellman. Le principe de programmation dynamique (PPD) nous amène à considérer la fonction valeur suivante

$$v(Q_t, B_t) = \sup_{b_s,\, s\in[t,(t+h)\wedge\tau]} \mathbb{E}\left[v(Q_{t+h}, B_{t+h}) - \int_t^{t+h} c(b_s)ds \,\middle|\, \mathscr{F}_t\right] \tag{1.7}$$

que nous supposerons suffisamment régulière (nous référons à [Kar81, Pha09] pour des études approfondies sur le contrôle stochastique et le PPD). Remarquons que les valeurs 0 et 1 sont des valeurs absorbantes pour Q, ainsi la fonction v devrait également satisfaire les conditions suivantes, pour tout $B \in [0, B_{MAX}]$

$$\begin{aligned} v(0^+, B) &= U(0) \\ v(1^-, B) &= U(1) \end{aligned}$$

ainsi que la condition au bord

$$v(Q, B_{\text{MAX}}) = U(Q) \quad \text{for} \quad Q \in (0,1) \tag{1.8}$$

Appliquant le principe de programmation dynamique, le processus $V_t = v(Q_t, B_t)$ est une sur-martinagle pour tout contrôle admissible, et une martingale pour le contrôle optimal. Ainsi, en appliquant la formule d'Itô pour un contrôle donné b lorsque $t + h \le \tau$.

$$\begin{aligned} & v(Q_{t+h}, B_{t+h}) \\ = \; & v(Q_t, B_t) + \int_t^{t+h} \frac{\partial v}{\partial Q}(Q_s, B_s)dQ_s + \int_t^{t+h} \frac{\partial v}{\partial B}(Q_s, B_s)dB_s + \frac{1}{2}\int_t^{t+h} \frac{\partial^2 v}{\partial Q^2}(Q_s, B_s)d\langle Q\rangle_s \end{aligned}$$

et en identifiant son drift nous établissons l'EDP de HJB (cf. preuve dans le chapitre 3): posons

$$A(B, b, Q) = \mu(B, b)Q(1-Q)\frac{\partial v}{\partial Q}(Q, B) + b\frac{\partial v}{\partial B}(Q, B) + \frac{1}{2}\sigma(B, b)^2 Q^2(1-Q)^2 \frac{\partial^2 v}{\partial Q^2}(Q, B) - c(b) \tag{1.9}$$

alors l'équation de HJB à l'intérieur du domaine $[0,1]\times[0,B_{MAX}]$ se réécrit

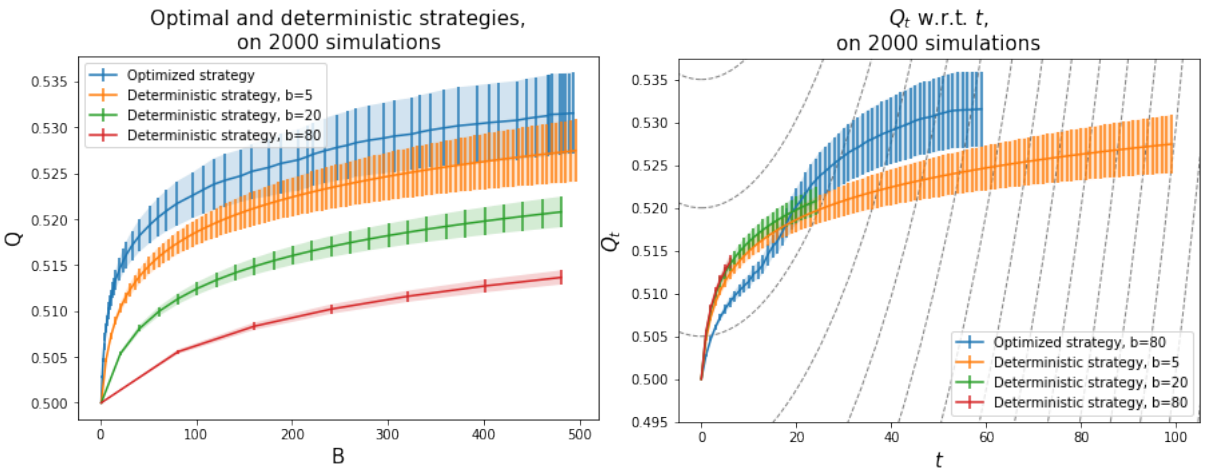
$$\sup_{b\geq 0, b\leq B_{MAX}-B} \{A(B,b,Q)\} = 0 \tag{1.10}$$

et le contrôle optimal b^* en fonction de (Q,B) est l'optimiseur de (1.10).

Quelques résultats numériques. Pour avoir la forme discrétisée de l'équation Eq.(1.10) nous devons discrétiser A (Eq.(1.9)). Pour ce faire, des méthodes d'approximation par différences finies peuvent être utilisées. La méthode des différences finies consiste à trouver une approximation de la solution d'une EDP aux "nœuds" d'une grille régulière. Ensuite, pour chaque (B,Q), nous pouvons utiliser l'*algorithme de Howard* [How60] pour déduire toutes les valeurs de $v_{i,j}$ par *induction rétrograde*. Cet algorithme consiste à alterner deux étapes jusqu'à un critère d'arrêt :

1. L'équation Eq. (1.10) sous forme discrétisée est résolue en remplaçant le contrôle optimal b précédemment calculé (ou initialisé) dans l'équation de Bellman discrétisée, ce qui nous donne une fonction de valeur v (candidate);

2. Le contrôle optimal (candidat) b est calculé, sur la base de la fonction valeur (candidate) v, en maximisant la forme discretisée de A.

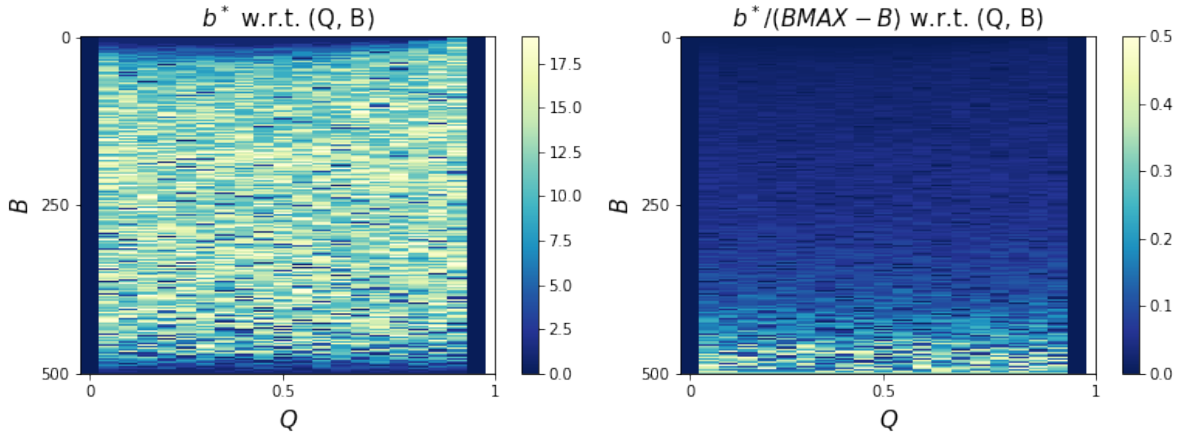


(a) (b)

Figure 1.6: Stratégies optimales et déterministes
Comparaison entre la stratégie optimisée et les stratégies déterministes avec la taille de lot statique $b \in \{5,20,80\}$. Initialement, nous fixons $(B_0,Q_0)=(0,0.5)$. Les résultats sont basés sur 2000 simulations et les zones colorées correspondent aux écarts types. (a) Affiche les trajectoires moyennes du processus d'état (B,Q) des stratégies optimisées et déterministes et (b) affiche la qualité moyenne en fonction du temps (ou itération) des stratégies optimisées et déterministes.

Strategy	$Q_0 = 0.3$	$Q_0 = 0.5$	$Q_0 = 0.7$	$Q_0 = 0.9$
Optimized	0.577 ± 0.01	0.756 ± 0.007	0.867 ± 0.001	0.959 ± 0.0
Deterministic with $b = 5$	0.574 ± 0.001	0.753 ± 0.011	0.867 ± 0.001	0.959 ± 0.0
Deterministic with $b = 20$	0.574 ± 0.0	0.728 ± 0.006	0.867 ± 0.003	0.959 ± 0.0
Deterministic with $b = 80$	0.574 ± 0.0	0.727 ± 0.0	0.842 ± 0.0	0.95 ± 0.01

Table 1.1: Fonctions valeur en fonction de la stratégie de contrôle. Nous reportons les moyennes et les écarts types sur les 2000 répétitions. Les valeurs colorées mettent en évidence la meilleure stratégie.



(a) (b)

Figure 1.7: Heatmaps de "X" par rapport au processus d'état (B, Q)
(a) X: le contrôle optimal b^*
(b) X: le taux du contrôle optimal dans le budget restant $b^*/(B_{\text{MAX}} - B)$

La Fig. 1.6 met en évidence les performances des stratégies optimisées en termes de temps et de qualité du modèle.
(1) La Fig. 1.6a montre que la stratégie optimisée est meilleure que les autres stratégies déterministes en étiquetant moins au début du processus pour rendre le modèle plus qualitatif avant de passer à des tailles de lots plus importantes (en accord avec les expériences BMAL qui seront plus explicitées dans la section 3.2 du chapitre 3). Les lecteurs intéressés peuvent se référer aux heatmaps de la Fig. 1.7 pour avoir un aperçu de la stratégie globale. Ce résultat est également visible sur les fonctions de valeur dans Tab. 1.1.
(2) Cette stratégie dynamique permet de réduire considérablement le nombre d'itérations. En effet, comme le montre la Fig. 1.6b, étant donné le budget B_{MAX} la stratégie dynamique prend moins de temps que la stratégie statique tout en étant plus qualitative.

Résumé des contributions. Nos principales contributions sont les suivantes :

1. Nous formulons le problème comme un problème de contrôle stochastique, que l'on résout

en utilisant le principe de programmation dynamique. Nous déterminons l'EDP de Hamilton–Jacobi–Bellman (HJB) que doit satisfaire la fonction valeur ([Kir04] pour plus de détails) et donnons une condition nécessaire et suffisante pour caractériser le contrôle optimal;

2. Nous étudions empiriquement le comportement du BMAL par la méthode de [WZS19], introduite dans 1.4, qui propose de requêter l'étiquette d'un lot d'instances (la taille de lot est fixée avant le requêtage) selon l'incertitude et la représentativité des instances (voir les détails de l'étude dans le chapitre 3);

3. Nous analysons numériquement le BMAL en termes de séquence optimale de tailles de lots et de recherche de stratégies d'étiquetage en pratique.

Des analyses numériques présentent une stratégie optimale d'étiquetage qui réduit considérablement le nombre d'itérations d'AL tout en gardant une bonne performance du modèle. Cela permettrait d'améliorer les conditions d'étiquetage pour les experts humains.

(Pré-)publications. Les chapitres 2 et 3 et ont donné lieu aux travaux suivants:

- Le chapitre 2 est le fruit d'un travail conjoint avec Romuald ELIE, Caroline HILLAIRET et Marc JUILLARD qui a donné lieu a une prépublication [EHHJ21] de titre "An overview of active learning methods for insurance with fairness appreciation." (Lien arxiv: `https://arxiv.org/abs/2112.09466`).

- Ce chapitre 3 est le fruit d'un travail conjoint avec Romuald ELIE et Caroline HILLAIRET qui devrait donné lieu à une pré-publication très prochainement.[21].

1.3.7 Perspective

Nous présentons ici les futures recherches que nous souhaitons mener sur le thème de l'apprentissage actif en assurance, qui semble être une suite naturelle de nos travaux de recherche.

L'étude sur le BMAL à taille variable met en évidence l'importance du choix de la séquence des tailles de batch. Nous avons formulé ce problème d'optimisation de manière dynamique. Cependant, à l'heure actuelle, cette méthodologie semble compliquée à utiliser en pratique sur de nouvelles bases de données. Une méthodologie consisterait à utiliser l'apprentissage par renforcement pour apprendre la politique des tailles optimales de batch à partir des données. Cette politique peut ensuite être généralisée à d'autres bases de données, ce qui la rend "réutilisable". L'objectif est de pouvoir transférer cette politique à d'autres procédures BMAL, réduisant ainsi le coût d'étiquetage et le coût d'attente tout en conservant une bonne performance du modèle de Machine Learning. Cette approche est assez proche du domaine d'apprentissage par transfert (ou *transfer learning* en anglais) [TS10] qui est l'amélioration de l'apprentissage d'une nouvelle tâche par le transfert des connaissances d'une

[21] Work in progress.

tâche connexe déjà apprise (nous nous référons à [WKW16] pour un aperçu des diverses méthodes). Cependant au lieu de transférer un modèle d'apprentissage nous voulons transférer une politique de séquence des tailles de batch dans les BMAL.

1.4 Équité algorithmique : état de l'art et contributions

Présentons l'état de l'art de l'équité algorithmique ainsi que les contributions de cette thèse dans ce domaine. L'équité algorithmique est devenue une préoccupation très importante au cours de la dernière décennie [ZWS+13, LJ16, CKP09, ZVGRG17, ADW19, ABD+18a, DOBD+18, CDH+19, CJS+20, BHN18]. En effet, elle s'intéresse à un problème social important : l'atténuation des biais historiques contenus dans les données. Il s'agit d'un problème crucial dans de nombreuses applications telles que l'évaluation des prêts, les soins de santé ou même les condamnations pénales. Pour rappel (voir 1.1.3), l'objectif commun de l'équité algorithmique est de réduire l'influence d'un attribut sensible sur une prédiction.

1.4.1 Notions d'équité algorithmique

Plusieurs notions d'équité de groupe ont déjà été considérées dans la littérature pour le problème de la classification binaire [ZVGRG19, BHN18]. Toutes imposent une condition d'indépendance entre la caractéristique sensible et la prédiction. Formellement, nous considérons $\mathscr{S} = \{-1, +1\}$ l'espace de la caractéristique sensible[22] (par exemple, le genre ou la religion). Nous considérons également $\mathscr{X}_{\setminus S} := \mathscr{X} \setminus \mathscr{S}$ l'espace des instances excluant l'espace de la caractéristique sensible, K le nombre total de classes et Y la vraie classe.

Présentons formellement les trois notions d'équité de groupe présentées (informellement) dans la section 1.1.3.

Definition 1.1: Égalité des chances (Equalized Odds)

Dans le cadre de l'Égalité des chances (voir [HPS16]), on dit qu'un classifieur $h \in \mathscr{H}$ est équitable par rapport à la distribution $\mathbb{P}$ sur $\mathscr{X}_{\setminus S} \times \mathscr{S} \times [K]$ si $h(X)$ et S sont indépendants conditionnellement à Y : $h(X) \perp\!\!\!\perp S \mid Y$ (cette notion aussi appelée *séparation*, voir [BHN17]). Ce qui revient à vérifier

$$\mathbb{P}(h(X) = 1 \mid S = 1, Y = k) = \mathbb{P}(h(X) = 1 \mid S = -1, Y = k), \ \forall k \in [K].$$

Pour une classification binaire, cette définition indique que le groupe protégé et le groupe non protégé doivent avoir des taux de faux positifs (ou *erreur de type I*) et des taux de vrais positifs égaux.

[22] Une caractéristique sensible multi-groupe peut être facilement généralisée.

Cette notion semble convenir lorsque la variable cible Y est une vérité-terrain (*groundtruth*) objective. Notons qu'un assouplissement de l'égalité des chances est possible dans le cas binaire : nous pouvons exiger que le groupe de non-discrimination ne soit présent que dans le résultat "favorisé" (par exemple, les admissions à l'université ou le recrutement d'employés). Cette justice est appelée Égalité des opportunités (*Equal Opportunity* en anglais) :

Definition 1.2: Égalité des opportunités (Equal Opportunity)

Dans un cadre binaire, nous disons qu'un classifieur $h \in \mathscr{H}$ satisfait l'égalité des opportunités (voir [HPS16]) par rapport à la distribution $\mathbb{P}$ sur $\mathscr{X}_{-S} \times \mathscr{S} \times \{0,1\}$ si

$$\mathbb{P}(h(X) = 1 \mid S = 1, Y = 1) = \mathbb{P}(h(X) = 1 \mid S = -1, Y = 1)$$

Présentons une autre notion d'équité : la Parité Démographique (*Demographic Parity* en anglais) qui exige d'avoir un groupe non-discriminatoire dans tous les résultats (prédits).

Definition 1.3: Parité Démographique (Demographic Parity)

Dans le cadre de la Parité Démographique (voir [CKP09]), on dit qu'un classifieur $h \in \mathscr{H}$ est équitable (ou juste) par rapport à la distribution $\mathbb{P}$ sur $\mathscr{X}_{-S} \times \mathscr{S} \times [K]$ si $h(X)$ et S sont indépendants : $h(X) \perp\!\!\!\perp S$ (cette notion est aussi appelée *indépendance*, voir [BHN17]). En d'autres termes,

$$\mathbb{P}(h(X) = k | S = 1) = \mathbb{P}(h(X) = k | S = -1)\,, \quad \forall k \in [K]\ .$$

Cette définition stipule que les groupes protégés et non protégés doivent avoir la même probabilité. Dans une version approximative, avec un seuil de tolérance $\epsilon > 0$ petit donné, nous voulons que

$$|\mathbb{P}(h(X) = k | S = 1) - \mathbb{P}(h(X) = k | S = -1)| \leq \epsilon\,, \quad \forall k \in [K]. \tag{1.11}$$

Nous parlons alors de ϵ-équitable (ou ϵ-fairness en anglais). Par contraste, nous appelons un classifieur *exact-équitable* (ou *exact-fairness*) si, dans le cadre DP, il est équitable (ou 0-équitable) avec la contrainte $\epsilon = 0$.

Dans ce document, nous nous concentrons sur la Parité Démographique (DP). La DP a un intérêt reconnu dans diverses applications, comme pour l'accord de prêt sans attributs de genre ou pour la prédiction de crimes sans discrimination ethnique [HDFMB11, KZC13, BS14, FFM+15]. En effet, lorsque nous ne pouvons pas faire confiance à l'objectivité de la variable cible Y alors la notion de DP est plus cohérente que les notions de séparation (notamment pour les préjugés sociaux et historiques).

Évaluation en parité démographique. Étant donné un classificateur $h \in \mathscr{H}$, lorsque l'équité est requise, deux aspects importants du classificateur doivent être contrôlés : son risque de mauvaise

classification $\mathscr{R}$ et son injustice évaluée par

$$\mathscr{U}(h) := \frac{1}{K}\sum_{k=1}^{K} |\mathbb{P}(h(X)=k|S=1) - \mathbb{P}(h(X)=k|S=-1)|\,.$$

Naturellement, compte tenu de la définition ci-dessus, un classifieur h est d'autant plus juste que $\mathscr{U}(h)$ devient petit. Un classifieur h est dit exact-équitable (resp. ε-équitable) si et seulement si $\mathscr{U}(h)=0$ (resp. $\mathscr{U}(h)\leq\varepsilon$). Cette quantité étant incalculable, sa forme empirique sera utilisée en pratique : pour des raisons de simplicité, nous notons $\hat{\nu}_{h|s}(k) = \frac{1}{|\mathscr{T}^s|}\sum_{(X,Y)\in\mathscr{T}^s} \mathbb{1}_{\{h(X)=k\}}$ la distribution empirique de $h(X)|S=s$ on $\mathscr{T}^s = \{(X,Y)\in\mathscr{D}^{(test)}|S=s\}$ l'ensemble des données test. L'inéquité est formellement définie par :

$$\hat{\mathscr{U}}(h) = \frac{1}{K}\sum_{k=1}^{K} \left|\hat{\nu}_{h|-1}(k) - \hat{\nu}_{h|1}(k)\right|.$$

1.4.2 Equité en Assurance

En assurance, l'équité algorithmique est une question importante, notamment pour la tarification des contrats d'assurance. Cette tarification repose sur la classification (ou segmentation) des risques, qui peut être basée sur des modèles de Machine Learning. Juridiquement chaque état, chaque régulateur fixe ses propres règles concernant les variables (*features*) de classification que les compagnies d'assurance peuvent ou ne peuvent pas utiliser lors de la tarification.

Prenons l'exemple de la tarification automobile aux États-Unis: certains États autorisent les assureurs à tenir compte de certaines variables personnelles, non liées à la conduite[23] (e.g. les antécédents en matière de crédit, l'emploi ou le genre) lors de la tarification automobile, tandis que d'autres États interdisent l'utilisation de ces variables, les considérant comme des critères injustes et discriminatoires (Nous renvoyons à [BC22] pour une discussion sur les variables protégées en assurance). Il est donc important de s'assurer que les algorithmes de tarification proposés n'utilisent pas in fine ces critères discriminatoires. Pour ce faire, différentes méthodologies peuvent être proposées.

1.4.3 Méthodologies

Il y a principalement trois façons d'établir des prédictions équitables : (1) les méthodes **pre-processing** qui atténuent les biais dans les données ; (2) les méthodes **in-processing** qui réduisent les biais pendant l'entraînement des algorithmes d'apprentissage et (3) les méthodes **post-processing** qui appliquent l'équité après l'ajustement des algorithmes d'apprentissage.

Méthode pre-processing. Les approches pre-processing tentent de transformer les données afin de supprimer la discrimination sous-jacente. Si l'algorithme est autorisé à modifier les données d'apprentissage, alors le pre-processing peut être utilisé. Par exemple, des travaux ont été réalisés sur les méthodes d'échantillonnage qui visent à corriger les données d'entraînement et à éliminer les

[23] https://www.thezebra.com/resources/research/car-insurance-rating-factors-by-state/

biais [CKS+18, DIKL18, HV19]. En particulier, [CKS+18] propose une stratégie d'*échantillonnage* qui échantillonne dans un grand ensemble de données (étiquetées) les instances à la fois diversifiées dans leurs caractéristiques et justes pour les attributs sensibles. Une autre stratégie est le *ré-étiquetage* des instances qui vise à modifier les étiquettes des données d'entraînement (et parfois de test) telles que la proportion des instances positives sont égales pour tous les groupes protégés [LRT11, CT17]. Enfin, nous pouvons *repondérer* les données d'apprentissage de sorte à attribuer des poids aux instances tout en laissant les données elles-mêmes inchangées (voir par exemple [KC12]).

Méthode in-processing. Les approches in-processing tentent de modifier et de changer les algorithmes de ML pour éliminer la discrimination pendant le processus d'apprentissage du modèle. S'il est permis de changer la procédure d'apprentissage d'un modèle ML, alors l'in-processing peut être utilisé pendant l'apprentissage d'un modèle, soit en incorporant des changements dans la fonction objectif [GCGF16, ABD+18b, Nar18], soit en imposant une contrainte [KAAS12, GYF18].

Méthode post-processing. Les approches post-processing tendent à appliquer des transformations aux inférences d'un modèle ML pour améliorer l'équité des prédictions [DHP+12, HPS16, KGZ19]. Le post-processing est l'une des approches les plus flexibles, car il ne nécessite qu'un accès aux prédictions et aux attributs sensibles, sans avoir besoin d'accéder aux algorithmes. Souvent, la méthodologie générale des algorithmes de post-processing consiste à prendre un sous-ensemble d'échantillons et à modifier leurs étiquettes prédites de manière appropriée afin de répondre à une exigence d'équité de groupe [KKZ12, HPS16, PRW+17]. En particulier, [KKZ12] propose de modifier les instances les plus incertaines (celles qui se trouvent dans la région d'incertitude du modèle ML), capturant ainsi les doutes humains sur les groupes non privilégiés. [CDH+19] présente des classifieurs binaires équitables sous les contraintes d'égalité des chances basées sur un seuil dépendant du groupe. La méthode post-processing présente plusieurs avantages en pratique, notamment, elle ne nécessite pas la modification des données et est agnostique aux modèles ML. Par conséquent elle est (1) adaptée aux environnements d'exécution et (2) elle est applicable à des scénarios de boîte noire où l'ensemble du pipeline ML n'est pas exposé.

Toutes les références mentionnées précédemment se concentrent principalement sur les cadres de classification binaire (pour rappel nous n'abordons pas la régression, pour un aperçu des méthodologies en régression voir [HM19]). Cependant, de nombreuses applications (modernes) relèvent de la classification multi-classes, par exemple la reconnaissance d'images ou la catégorisation de textes. De plus, la plupart des applications du monde réel peuvent être abordées dans une perspective multi-classes. Par exemple, la récidive criminelle est souvent traitée comme un problème binaire, alors qu'il peut être plus approprié de distinguer les différentes strates du problème et de fournir une description plus fine du comportement criminel. Cependant, l'extension des travaux précédents au cadre multi-classes est délicate, en particulier parce que la notion adéquate d'équité dans ce cadre n'est pas clairement définie et doit être traitée avec prudence.

1.4.4 Équité algorithmique dans la classification multi-classes

À notre connaissance, peu de travaux sur l'équité dans la classification multi-classes ont vu le jour[24]. Ces méthodes sont les suivantes :

Classification multi-classes équitable basée sur les SVM. En classification multi-classes, les auteurs du papier [YX20] se concentrent sur la prédiction équitable des machines à vecteurs de support (SVM) : leur approche de l'équité repose sur la sélection correcte d'un sous-ensemble de données qui n'est pas biaisé du point de vue de l'équité. En outre, la procédure décrite dans [YX20] choisit d'imposer l'équité à chaque composante de la fonction de score.

Variante de classification multi-classes équitable : *score-fair*. Nous proposons ici une autre méthode d'équité dans la classification multi-classes que nous considérons comme une approche alternative et post-processing de [YX20] : le *score-fair*. Pour une fonction de score (mesurable) donnée $h(\cdot) = (h_1(\cdot), \dots, h_K(\cdot))$ mettant en correspondance $\mathscr{X}$ avec $\mathbb{R}^K$, l'approche *score-fair* consiste à imposer l'équité au niveau de chaque fonction de score en utilisant la totalité des données étiquetées. Une façon possible d'aborder ce problème est de considérer la tâche de minimisation suivante

$$h^*_{\text{score-fair}} \in \operatorname{argmin} \left\{ R_2(h) : h \text{ est } \textit{score-fair} \right\} \quad . \tag{1.12}$$

où $R_2(h) = \mathbb{E}\left[\sum_{k=1}^K (Y_k - h_k(X))^2\right]$ est le risque L_2. h est *score-fair* signifie que pour tout $k \in [K]$ et pour tout $t \in \mathbb{R}$ nous avons

$$\mathbb{P}(h_k(X) \leq t \mid S = -1) = \mathbb{P}(h_k(X) \leq t \mid S = 1) \quad .$$

La solution optimale du problème (1.12) est séparable et peut alors être résolue élément par élément.

Bien que cette approche semble plutôt naturelle, soulignons que *score-fair* DP n'implique pas DP pour le maximiseur de score, puisque l'opération de maximisation, contrairement au seuillage, ne préserve pas la propriété DP. Les fonctions optimales $(h^*_{\text{score-fair}})_k$ de *score-fair* reposent sur le risque L_2 et peuvent être facilement caractérisées en suivant l'approche dans [GLR20, CDH+20c] qui renforce l'équité en utilisant les barycentres de Wasserstein. Cette méthodologie d'équité, conçue à l'origine pour les problèmes de régression, peut être facilement adaptée aux problèmes multi-classes. En particulier Thm. 2.3 du papier [CDH+20c] identifie la distribution du classifieur score-fair $h^*_{score-fair}$ en tant que solutions d'un problème de barycentre de Wasserstein. Pour les lecteurs intéressés, dans le chapitre 5 la procédure d'estimation de $h^*_{score-fair}$ est décrite par Alg. 6 avant de proposer plusieurs expérimentations numériques.

Enfin en DP, au lieu d'imposer l'équité sur toutes les fonctions de score, nous proposon dans [DEHH21] une procédure qui impose l'équité post-processing directement sur le score maximal (nous utilisons ce score car il est associé au classifieur de Bayes, c'est-à-dire le classifieur optimal pour un

[24] le jour où ce manuscrit est écrit, début 2022.

problème de classification donné). Cette méthodologie, qui est brièvement décrite dans la section 1.4.5, fait l'objet du chapitre 4.

1.4.5 Contributions

Présentons à présent une vue d'ensemble des principales contributions du chapitre 4 et du chapitre 5 à l'équité algorithmique.

1.4.5.1 Garantie d'équité dans la classification multi-classes.

Contexte. Comme précisé ci-avant, dans la littérature, l'imposition d'une contrainte d'équité dans le problème multi-classes n'a été brièvement abordée que dans (1) [YX20] et (2) l'approche alternative réadaptée *score-fair*. Ces procédures choisissent d'imposer l'équité à chaque composante de la fonction de score. Cependant, étant donné que la règle de décision dans le cadre multi-classes repose sur le maximiseur portant sur les scores, nous n'adoptons pas cette approche assez peu naturelle et imposons plutôt directement l'équité sur le maximiseur lui-même.

Soit (X,S,Y) un n-uplet aléatoire avec la distribution $\mathbb{P}$, où $X \in \mathscr{X}$ un sous-ensemble de $\mathbb{R}^d$, $S \in \mathscr{S} := \{-1,1\}$, et $Y \in [K] := \{1,\dots,K\}$ avec K étant un nombre fixe de classes. La distribution de la caractéristique sensible S est désignée par $(\pi_s)_{s\in\mathscr{S}}$, et nous supposons que $\min_{s\in\mathscr{S}} \pi_s > 0$. Une règle de classification g est une fonction faisant correspondre $\mathscr{X} \times \{-1,1\}$ sur $[K]$, dont la performance est évaluée par le risque de mauvaise classification

$$\mathscr{R}(g) := \mathbb{P}(g(X,S) \neq Y) \quad .$$

Pour $k \in [K]$, nous désignons $p_k(X,S) := \mathbb{P}(Y = k|X,S)$. Rappelons qu'un classifieur de Bayes minimisant le risque de mauvaise classification $\mathscr{R}(\cdot)$ sur l'ensemble $\mathscr{G}$ de tous les classifieurs est donné par

$$g^*(x,s) \in \arg\max_k p_k(x,s), \quad \text{pour tout } (x,s) \in \mathscr{X} \times \mathscr{S} \quad .$$

Nous étendons les deux définitions de l'équité exacte et approximative dans le cas de la *Parité démographique* à la classification multi-classes.

L'équité exacte dans la classification multi-classes. Fournissons une formulation explicite des classifieurs exact-équitables optimaux par rapport au risque de mauvaise classification sous la contrainte DP. Un classifieur exact-équitable optimal g^*_{fair} résout

$$\min_{g\in\mathscr{G}_{fair}} \mathscr{R}(g) \quad .$$

L'obtention d'un classificateur équitable optimal nécessite d'équilibrer correctement le risque de mauvaise classification et le critère d'équité. Dans ce but, considérons le Lagrangien du problème ci-dessus et introduisons pour $\lambda = (\lambda_1,\dots,\lambda_K) \in \mathscr{R}^K$,

$$\mathscr{R}_\lambda(g) := \mathscr{R}(g) + \sum_{k=1}^{K} \lambda_k [\mathbb{P}(g(X,S) = k|S = 1) - \mathbb{P}(g(X,S) = k|S = -1)] \quad . \tag{1.13}$$

Nous appelons cette mesure *fair-risk* et nous détaillons ci-dessous comment la minimisation de ce risque pour un λ correctement choisi donne le classifieur équitable g^*_{fair}.

Proposition 1.1

Supposons que la fonction $t \mapsto \mathbb{P}\left(p_k(X,S) - p_j(X,S) \leq t | S = s\right)$ est continue, pour tout $k, j \in [K]$ et $s \in \mathcal{S}$. Définissons

$$\lambda^* \in \arg\min_{\lambda \in \mathbb{R}^K} \sum_{s \in \mathcal{S}} \mathbb{E}_{X|S=s}\left[\max_k (\pi_s p_k(X,s) - s\lambda_k)\right] \quad .$$

Alors, $g^*_{fair} \in \arg\min_{g \in \mathcal{G}_{fair}} \mathcal{R}(g)$ si et seulement si $g^*_{fair} \in \arg\min_{g \in \mathcal{G}} \mathcal{R}_{\lambda^*}(g)$.
Notons que l'hypothèse de continuité implique que la distribution des différences $p_k(X,S) - p_j(X,S)$ n'a pas d'atomes.

Par construction, $\mathcal{R}_{\lambda^*}$ est une mesure de risque qui équilibre efficacement à la fois la précision de la classification et l'équité.

Tout classifieur équitable optimal consiste simplement à maximiser les scores, qui sont obtenus en déplaçant les probabilités conditionnelles originales de manière optimale.

Corollary 1

Sous l'hypothèse de continuité définie ci-dessus, un classifieur exact équitable optimal est caractérisé

$$g^*_{fair}(x,s) \in \arg\max_k \left(\pi_s p_k(x,s) - s\lambda^*_k\right),\ (x,s) \in \mathcal{X} \times \mathcal{S}.$$

Ceci suggère une procédure *plug-in* pour laquelle nous établissons des garanties théoriques détaillées dans le chapitre 4. Il est prouvé que l'estimateur amélioré imite le comportement de la règle optimale à la fois en termes d'équité et de risque.

L'équité approximative dans la classification multi-classes. L'équité approximative, définie par l'équation 1.11 de la section 1.4.1, également appelée équité ε, est particulièrement populaire d'un point de vue pratique dans le domaine de l'équité algorithmique. Notons que l'utilisateur est autorisé à relâcher la contrainte d'équité chaque fois que cela est pertinent ou nécessaire. Une telle relaxation est cruciale lorsque l'équité exacte dégrade fortement la précision de la méthode. Notre objectif est de poser une formulation explicite d'un classifieur ε-équitable optimal par rapport au risque de mauvaise classification, dénoté par $g^*_{\varepsilon-fair}$, qui est la solution de

$$\min_{g \in \mathcal{G}_{\varepsilon-fair}} \mathcal{R}(g) \quad .$$

La résolution de ce problème présente des similitudes avec le cas de l'équité exacte. La première étape est d'écrire le Lagrangien du problème ci-dessus : pour $\lambda^{(1)} = (\lambda_1^{(1)}, \dots, \lambda_K^{(1)}) \in \mathbb{R}_+^K$ et $\lambda^{(2)} = (\lambda_1^{(2)}, \dots, \lambda_K^{(2)}) \in \mathbb{R}_+^K$ nous définissons le *ε-fair-risk* comme suit

$$\begin{aligned} \mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g) := \mathscr{R}(g) &+ \sum_{k=1}^{K} \lambda_k^{(1)} [\mathbb{P}(g(X,S)=k|S=1) - \mathbb{P}(g(X,S)=k|S=-1) - \varepsilon] \\ &+ \sum_{k=1}^{K} \lambda_k^{(2)} [\mathbb{P}(g(X,S)=k|S=-1) - \mathbb{P}(g(X,S)=k|S=1) - \varepsilon] \ . \end{aligned} \tag{1.14}$$

Un analogue de la Proposition 1.1 suit ainsi qu'une caractérisation complète du classificateur ε-fair optimal.

Proposition 1.2

Posons $H : \mathbb{R}_+^{2K} \to \mathbb{R}$ la fonction

$$H(\lambda^{(1)}, \lambda^{(2)}) \quad = \quad \sum_{s \in \mathscr{S}} \mathbb{E}_{X|S=s} \left[\max_k \left(\pi_s p_k(X,s) - s(\lambda_k^{(1)} - \lambda_k^{(2)}) \right) \right] \ + \ \varepsilon \sum_{k=1}^{K} (\lambda_k^{(1)} \ + \ \lambda_k^{(2)}) \ .$$

Supposons que l'hypothèse de continuité de de la proposition 1.1 est satisfaite et définissons $\lambda^{*(1)}, \lambda^{*(2)} \in \mathbb{R}_+^{2K}$ par

$$(\lambda^{*(1)}, \lambda^{*(2)}) \in \arg \min_{(\lambda^{(1)},\lambda^{(2)}) \in \mathbb{R}_+^{2K}} H(\lambda^{(1)}, \lambda^{(2)}) \ .$$

Alors, $g^*_{\varepsilon-fair} \in \arg\min_{g \in \mathscr{G}_{\varepsilon-fair}} \mathscr{R}(g)$ si et seulement si $g^*_{\varepsilon-fair} \in \arg\min_{g \in \mathscr{G}} \mathscr{R}_{\lambda^{*(1)},\lambda^{*(2)}}(g)$.
De plus, pour tout $(x,s) \in \mathscr{X} \times \mathscr{S}$, nous pouvons réécrire

$$g^*_{\varepsilon-fair}(x,s) = \arg\max_{k \in [K]} \left(\pi_s p_k(x,s) - s(\lambda_k^{*(1)} - \lambda_k^{*(2)}) \right).$$

L'approche semi-supervisée. La procédure d'estimation améliorée se déroule en deux étapes. Nous construisons d'abord des estimateurs des probabilités conditionnelles $(p_k)_k$. Ensuite, on considère l'estimation des paramètres $\lambda^{*(1)}$, $\lambda^{*(2)}$ et $(\pi_s)_{s \in \mathscr{S}}$. Plus précisément, notre procédure basée sur les données est semi-supervisée car elle s'appuie sur deux ensembles de données indépendants, l'un étiqueté $\mathscr{D}_n = (X_i, S_i, Y_i)_{i=1,\dots,n}$ et l'autre non étiqueté $\mathscr{D}'_N$ contient N (des copies *i.i.d.* de (X,S)).

- L'ensemble des données *étiquetées* permet d'entraîner des estimateurs $(\hat{p}_k)_k$ des probabilités conditionnelles $(p_k)_k$ (*e.g.*, Random Forest ou SVM).
- L'ensemble des données *non étiquetées* est utilisé pour estimer des quantités telles que les distributions marginales. Par conséquent, l'échantillon $(S_1, \dots, S_N)$ de caractéristiques sensibles est utilisé pour calculer les fréquences empiriques $(\hat{\pi}_s)_{s \in \mathscr{S}}$ en tant qu'estimations de $(\pi_s)_{s \in \mathscr{S}}$. Pour $s \in \mathscr{S}$, le nombre d'observations correspondant à $S = s$ est noté N_s, de sorte que $N_{-1} + N_1 = N$. Les vecteurs caractéristiques associés dans $\mathscr{D}'_N$ sont notés $X_1^s, \dots, X_{N_s}^s$.

L'analyse théorique du risque et de l'injustice de la règle du plug-in nécessite des conditions de continuité sur les variables aléatoires $\hat{p}_k(X,S)$. Cependant, cette propriété est automatiquement satisfaite chaque fois que l'on perturbe $(\hat{p}_k)_k$ avec un "petit" bruit aléatoire. Pour être plus précis, nous introduisons $\bar{p}_k(X,S,\zeta_k) := \hat{p}_k(X,S) + \zeta_k$, pour une perturbation uniforme donnée ζ_k sur $[0,u]$. Nous posons par la suite $(\zeta_k)_{k\in[K]}$ et $(\zeta^s_{k,i})$ des copies indépendantes d'une distribution uniforme sur $[0,u]$.

L'objectif est d'estimer toutes les quantités inconnues et de les insérer dans l'expression de $g^*_{\varepsilon-fair}$. Cela permet d'écrire l'estimateur plug-in

$$\hat{g}_\varepsilon(x,s) = \arg\max_{k\in[K]} \left(\hat{\pi}_s \bar{p}_k(x,s,\zeta_k) - s(\hat{\lambda}^{(1)}_k - \hat{\lambda}^{(2)}_k) \right) \quad , \tag{1.15}$$

pour tout $(x,s) \in \mathscr{X} \times \mathscr{S}$, où le couple $(\hat{\lambda}^{(1)}, \hat{\lambda}^{(2)})$ est le minimiseur sur $\mathbb{R}^{2K}_+$ de $\hat{H}(\lambda^{(1)}, \lambda^{(2)})$ qui est défini comme suit

$$\hat{H}(\lambda^{(1)}, \lambda^{(2)}) \quad = \quad \sum_{s\in\mathscr{S}} \frac{1}{N_s} \sum_{i=1}^{N_s} \left[\max_k \left(\hat{\pi}_s \bar{p}_k(X^s_i, s, \zeta^s_{k,i}) - s(\lambda^{(1)}_k - \lambda^{(2)}_k) \right) \right] \; + \; \varepsilon \sum_{k=1}^{K} (\lambda^{(1)}_k \; + \; \lambda^{(2)}_k) \; . \tag{1.16}$$

Quelques garanties théoriques. Nous fournissons maintenant la consistance de $\hat{g}_\varepsilon$ par rapport au risque de mauvaise classification. Nous définissons la norme L_1 dans $\mathbb{R}^K$ entre l'estimateur $\hat{\mathbf{p}} := (\hat{p}_1, \ldots, \hat{p}_K)$ et le vecteur des probabilités conditionnelles $\mathbf{p} := (p_1, \ldots, p_K)$ par $\|\hat{\mathbf{p}} - \mathbf{p}\|_1 = \sum_{k\in[K]} |\hat{p}_k(X,S) - p_k(X,S)|$. Nous suggérons une procédure *plug-in* pour laquelle nous établissons des garanties théoriques:

i) *ε-équitable non paramétrique.* Pour tout estimateur $\hat{p}_k$, l'estimateur $\hat{g}_\varepsilon$ atteint le bon niveau d'équité, à savoir, $|\mathbb{E}[\mathscr{U}(\hat{g}_\varepsilon)] - \varepsilon| \leq \frac{C}{\sqrt{N}}$ pour une certaine constante positive C.

ii) *Résultats de consistance.* Si l'estimateur initial des probabilités conditionnelles garantit la consistance en norme L_1, c'est-à-dire si $\mathbb{E}\left[\|\hat{\mathbf{p}} - \mathbf{p}\|_1\right] \to 0$ et si $u = u_n \to 0$ quand $n \to \infty$, alors $\mathbb{E}[\mathscr{R}(\hat{g}_\varepsilon)] \to \mathscr{R}(g^*_{\varepsilon-fair})$ quand $n, N \to \infty$.

Nous obtenons ainsi de solides garanties théoriques pour $\hat{g}_\varepsilon$ en termes d'équité et de risque. Notre approche permet de contrôler spécifiquement l'équité de l'algorithme et de la fixer à un niveau souhaité.

Résultats numériques. Les approches exact-équitables et ϵ-équitables évaluées sur des ensembles de données synthétiques et réelles s'avèrent très efficaces dans la prise de décision avec un niveau d'injustice prédéfini.

- *Évaluation sur les données synthétiques.* La Fig. 1.8-Gauche illustre l'équité et la précision de notre algorithme pour différents niveaux de biais historique quantifié par p dans l'ensemble de données synthétiques. Plus précisément, le modèle devient équitable lorsque $p = 0,5$ et complètement injuste lorsque $p \in \{0,1\}$ (voir section 4.6.1 du chapitre 4 pour le schéma de simulation). D'après la Fig. 1.8-Gauche, nous remarquons que 1) l'efficacité de l'algorithme en matière d'équité est particulièrement significative pour les jeux de données présentant un biais historique important ($p = 0,95$ ou $0,99$) ; 2) notre méthode parvient à atteindre le niveau d'injustice demandé jusqu'à

de petits termes d'approximation (voir comment les courbes sont verticales dès que la limite de l'injustice est atteinte) et 3) l'efficacité en matière d'équité est contrebalancée par une plus faible précision (nous contrôlons le compromis entre l'injustice et la précision par le paramètre ε).

- *Application aux données réelles.* Les résultats dans le cadre multi-classes sont présentés dans le tableau 1.2 et soulignent l'efficacité de notre méthode *argmax-fair* (ou *exact-fairness*) (voir section 5.5 du chapitre 5 pour le schéma de simulation). Par exemple, pour le jeu de données LAW et le GaussSVC avec *argmax-fair*, l'injustice est divisée par presque 25 (0,97 à 0,04). En outre, la procédure *argmax-fair* surpasse les algorithmes *unfair* et *score-fair* pour les jeux de données CRIME, LAW et WINE en termes d'injustice. Cependant, nous observons une légère diminution de la précision des modèles (relativement faible par rapport au gain en équité). Notez que pour le jeu de données CMC, *score-fair* et *argmax-fair* obtiennent des performances similaires.

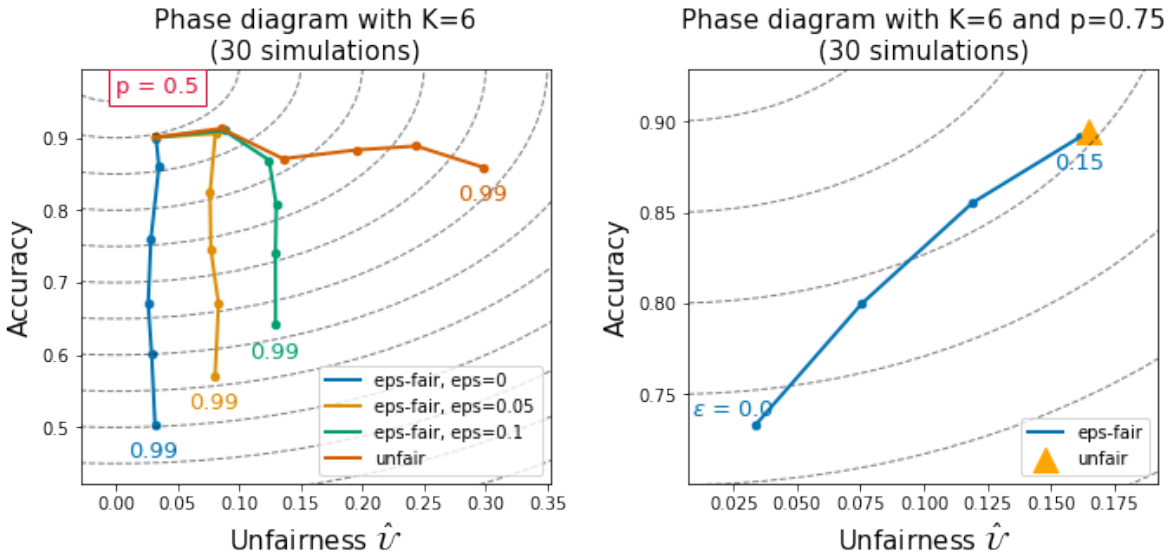


Figure 1.8: (Accuracy, Unfairness) diagrammes de phase pour les ensembles de données synthétiques. Simulations sur des données synthétiques à $K = 6$ classes. *Gauche* le niveau de biais p ; *Droite* le paramètre de compromis précision/équité ε. Le coin supérieur gauche donne le meilleur compromis (Accuracy, Unfairness).

Résumé des contributions. Nos principales contributions sont les suivantes :

- Nous étendons la notion de DP d'équité exacte et approximative au problème de la classification multi-classes ;

- Nous donnons des solutions optimales pour le problème du classificateur multi-classes sous des contraintes DP exactes ou approximatives ;

METHOD \ DATA	CRIME, K = 7		LAW, K = 4		WINE, K = 5		CMC, K = 3	
	Accuracy	Unfairness	Accuracy	Unfairness	Accuracy	Unfairness	Accuracy	Unfairness
reglog + unfair	0.34 ± 0.02	1.12 ± 0.07	0.43 ± 0.01	0.89 ± 0.05	0.54 ± 0.01	0.47 ± 0.05	0.52 ± 0.02	0.78 ± 0.16
reglog + score-fair (baseline)	0.33 ± 0.01	0.78 ± 0.09	0.42 ± 0.01	0.09 ± 0.02	0.54 ± 0.01	0.08 ± 0.03	0.51 ± 0.02	0.25 ± 0.1
reglog + argmax-fair	0.28 ± 0.01	0.26 ± 0.07	0.42 ± 0.01	0.05 ± 0.02	0.54 ± 0.02	0.04 ± 0.01	0.52 ± 0.02	0.19 ± 0.1
linearSVC + unfair	0.36 ± 0.02	1.12 ± 0.07	0.43 ± 0.01	0.97 ± 0.07	0.53 ± 0.01	0.27 ± 0.05	0.51 ± 0.02	0.63 ± 0.22
linearSVC + score-fair (baseline)	0.31 ± 0.02	0.88 ± 0.05	0.42 ± 0.01	0.1 ± 0.03	0.53 ± 0.01	0.1 ± 0.07	0.53 ± 0.02	0.26 ± 0.16
linearSVC + argmax-fair	0.29 ± 0.02	0.25 ± 0.08	0.42 ± 0.01	0.04 ± 0.02	0.53 ± 0.01	0.06 ± 0.04	0.52 ± 0.02	0.2 ± 0.12
GaussSVC + unfair	0.36 ± 0.02	1.4 ± 0.13	0.43 ± 0.01	1.04 ± 0.04	0.53 ± 0.01	0.28 ± 0.06	0.51 ± 0.02	1.0 ± 0.17
GaussSVC + score-fair (baseline)	0.35 ± 0.02	1.02 ± 0.07	0.42 ± 0.01	0.16 ± 0.04	0.55 ± 0.01	0.12 ± 0.04	0.51 ± 0.02	0.16 ± 0.09
GaussSVC + argmax-fair	0.3 ± 0.02	0.22 ± 0.05	0.42 ± 0.01	0.10 ± 0.03	0.55 ± 0.01	0.06 ± 0.03	0.5 ± 0.03	0.2 ± 0.08
RF + unfair	0.37 ± 0.02	1.02 ± 0.04	0.40 ± 0.01	0.65 ± 0.04	0.66 ± 0.01	0.31 ± 0.05	0.55 ± 0.02	0.35 ± 0.18
RF + score-fair (baseline)	0.34 ± 0.02	0.67 ± 0.06	0.39 ± 0.01	0.11 ± 0.05	0.66 ± 0.01	0.09 ± 0.03	0.52 ± 0.03	0.21 ± 0.08
RF + argmax-fair	0.3 ± 0.02	0.33 ± 0.11	0.39 ± 0.01	0.07 ± 0.02	0.66 ± 0.01	0.08 ± 0.02	0.55 ± 0.02	0.22 ± 0.13

Table 1.2: Performance (accuracy & unfairness) des méthodes pour tous les ensembles de données (avec K le nombre de classes) et classifieurs. Nous rapportons les moyennes et les écarts types sur les 30 répétitions. Les valeurs colorées soulignent la meilleure équité.

- Nous construisons une procédure basée sur les données qui imite la performance de la règle optimale à la fois en termes de risque et d'équité. Notamment, nos garanties d'équité sont *distribution-free* ;

- La robustesse de notre approche est illustrée sur des jeux de données synthétiques avec différents niveaux de biais, ainsi que sur plusieurs jeux de données réels. Elle s'avère très efficace pour la prise de décision avec un niveau prédéfini ε d'injustice (voir la notion de ε-équitable définie dans Eq. (1.11)).

- L'approche est évaluée sur des ensembles de données synthétiques et réelles et s'avère être compétitive par rapport à la méthodologie in-processing *fair-learn* dans le cas binaire [ABD+18b].

1.4.5.2 Comparaison de notre contrainte d'équité avec l'approche alternative *score-fair*.

Comme expliqué dans la section 1.4.4, dans le chapitre 5, nous mettons en évidence plusieurs aspects numériques de la procédure du classifieur exact-équitable présentée dans le chapitre 4. Comme point de repère, à la différence du chapitre 4, nous étudions l'approche alternative *score-fair* qui impose l'équité sur chaque score individuel qui est une variante de [YX20]. Ensuite, nous illustrons l'efficacité de notre procédure pour construire des prédictions justes et fiables sur des jeux de données synthétiques et réels. Notamment par rapport à l'approche *score-fair* l'efficacité de notre algorithme en termes d'équité est particulièrement remarquable pour les ensembles de données présentant un biais historique important.

(Pré-)publications. Les chapitres 4 et 5 sont le fruit d'un travail conjoint avec Christophe Denis, Romuald Elie et Mohamed Hebiri et ont donné lieu à une prépublication [DEHH21] de titre "Fairness guarantee in multi-class classification" (Lien arxiv: `https://arxiv.org/abs/2109.13642`).

1.4.6 Perspective

Sur le sujet de l'équité, les résultats que nous venons de présenter ouvrent la voie à de nombreuses perspectives:

Équité dans la classification multi-classes avec un attribut sensible continue. Nos études ne considèrent qu'une caractéristique sensible binaire (ou multigroupe) où S est fini (*e.g.* genre ou orentiation sexuel). Cependant, dans la pratique, le **cas continu** pourrait être tout aussi important, car de nombreuses entreprises s'efforcent d'être équitables et d'atténuer la discrimination en fonction de caractéristiques non finies comme *âge* ou *salaire*. Par rapport au cadre discret où la DP peut être réduite pour garantir que $\mathbb{E}[\hat{Y}|S] = \mathbb{E}[\hat{Y}]$ [ABD+18a], le cas continu est plus difficile puisque nous devons considérer les divergences de distribution au lieu des probabilités conditionnelles.

Équité dans la classification multi-labels. La classification multi-labels consiste à attribuer un ensemble d'étiquettes (au lieu d'une seule en classification multi-classes) cibles à chaque échantillon. Nous rappelons que l'approche *score-fair*, qui repose sur le risque L_2, impose l'équité sur chaque score individuel $(h^*_{\text{score-fair}})_k$. Bien que pour la classification multi-classes il semble naturel d'imposer l'équité sur le maximiseur (cf. méthodologies d'équité exacte ou approximative introduites dans la section 1.4.5.1 et développées dans le chapitre 4), l'approche *score-fair* semble être une contrainte d'équité naturelle pour la classification multi-labels.

1.5 Organisation du manuscrit

Cette thèse est organisée autour de trois principaux chapitres (Chap 2, 3 et 4) qui font l'objet de trois articles de recherche (présentées dans les sections 1.3.6 et 1.4.5). Enfin le chapitre 5 présente une étude complémentaire. En résumé:

- Le chapitre 2 présente les différentes méthodologies classiques d'apprentissage actif avant d'étudier leur impact empirique (en termes de précision, de robustesse et d'équité) sur des ensembles de données synthétiques et réelles ;
- Le chapitre 3 étudie l'apprentissage actif en mode batch et détermine la séquence de taille optimale de batch, en fonction d'un critère combinant qualité de prédiction et coût d'étiquetage ;
- Dans le chapitre 4 nous étendons les deux définitions de l'équité exacte et approximative, dans le cas de la parité démographique, à la classification multi-classes et spécifions les classifieurs équitables optimaux correspondants pour lesquelles nous établissons des garanties théoriques.
- le chapitre 5, qui est une extension du chapitre 4, propose de comparer les deux contraintes d'équité *score-fair* et *argmax-fair*.

Chapter 2

An overview of active learning methods for insurance with fairness in mind

Contents

This paper addresses and solves some challenges in the adoption of machine learning in insurance with the democratization of model deployment. The first challenge is reducing the labelling effort (hence focusing on the data quality) with the help of *active learning*, a feedback loop between the model inference and an oracle: as in insurance the unlabeled data is usually abundant, active learning can become a significant asset in reducing the labelling cost. For that purpose, this paper sketch out various classical active learning methodologies before studying their empirical impact on both synthetic and real datasets. Another key challenge in insurance is the *fairness* issue in model inferences. We will introduce and integrate a *post-processing* fairness for multi-class tasks in this active learning framework to solve these two issues. Finally numerical experiments on unfair datasets highlight that the proposed setup presents a good compromise between model precision and fairness.

Keywords: Active learning, algorithmic fairness, insurance datasets.

2.1 Introduction

Over the last decade, technological advances have allowed the emergence of Artificial Intelligence (AI) solutions. This emergence has been accompanied by a shift in AI research: as the capabilities of artificial intelligence become marketable, research is now mainly driven by companies and no longer by government funding. Thus, Facebook, Amazon, Microsoft and Google have largely participated in its democratization. This democratization has allowed many advances (autonomous cars, translation...) and facilitates the implementation and deployment of AI solutions.

New functionalities are regularly made possible thanks to the availability of new neural network structures that are pre-trained and rapidly integrated into high-level frameworks such as keras[1] in python. However, this appearance of simplicity hides in some cases a relatively complex prerequisite: having a large volume of training data. In order to understand this need it may be useful to distinguish between generic and specific AI solutions:

- A cognitive service can be said to be generic as soon as it is not dedicated to a sector of activity[2]. They are built, made available and updated by the Big Tech and do not require any adaptation before reuse. So when a company pays to use this service, it is paying for three main things: the mathematical performance of the underlying model, the technical performance of

[1] We can cite for example BERT, a language model developed by Google in 2018. This method has significantly improved performances in automatic language processing.

[2] For example, the Vision API of Azure (OCR technology) has been trained by Microsoft to extract printed text in several languages, handwritten text in English as well as numbers and monetary symbols from images or PDFs. Whatever the field of activity and under the assumption of not using a specific language, it is possible to use this service without retraining

the infrastructure hosting the solution, and the sheer volume of data and machine power that was required to train the model.

- At the opposite end of the spectrum from these solutions are the so-called specific cognitive services. By analogy, these services are so called because they are totally or partially dedicated to a specific sector of activity[3]. But unlike generic services, these solutions are rarely worked on by the big tech companies. Not because their specific character makes them more complex but rather because their niche status leads to a low Return On Investment (ROI), mainly due to the reduced scope of their use and because they require the intervention of rare and expensive business experts. As a result, it is difficult or even impossible to find pre-trained solutions that meet the company's needs, either free of charge or for a fee.

If the separation between specific and generic services seems relatively simple, it is illusory to think that the main AI business projects are based on a single category of service. The reality in the field is that the two approaches are mixed. As a result, most AI projects require large amounts of data to train specific algorithms, which in practice are often based on deep learning models and are extremely data-intensive. For example, for text classification (i.e. topic detection), the training set must contains at least 100 documents for each topic and the building of an insurance document parser will need a training set of at least 5000 documents. The labellisation step is obviously an important part of the AI process (and perhaps the most important). Indeed qualitative labeled data helps in calibrating the learning model to correctly map instances and labels and the lack of it can badly impact the performance of the model and may sometimes introduce biases and ethical problems (e.g. the fairness in AI solutions like racial problems in recruitment). This time consuming (and expensive) task has to be performed by experts in order to insure the quality of labels. Labeling campaigns are therefore often carried out under time (and therefore volume) constraints. Having algorithms to prioritize the data to be labeled instead of randomly selecting these examples is in some cases as important as the deep learning algorithm that will be trained with these data. These algorithms refer to the field of active learning. This field, which is a sub-section of machine learning, enables the learning algorithm to interactively query a human expert (a.k.a. *oracle*) to label new data points with the desired outputs.

From theory to practice the field of active learning has received a lot of attention in the recent years, such as [BGNK18, SED19, SNL+18] among others. Furthermore in the literature there exists many active learning surveys [Set09, FZL13, RXC+20]. However few works tackle the special challenges related to actuarial problems such as fairness and transparency. It is well-known that enforcing fairness in the model drop the model accuracy as shown by several experimental results [ZVGRG19, ABD+18a, DOBD+18, CDH+19, BHN18]. Nevertheless, fairness in AI is one of the key challenges of insurers (notably for EU insurers with the development of regulations such as GDPR) and the fairness evaluation

[3]For example, let us consider an entity wishing to measure compliance with GDPR rules in the context of calls made on its platforms. If the model to be implemented is based on generic speech-to-text algorithms, the notion of "compliant" or "non-compliant" verbatims in the GDPR and insurance sense is specific to the sector of activity (or even to the company in the case where its "jargon" is strong)

of active learning methods are under-studied. Up to our knowledge only three papers investigate the issue of fairness in active learning. [AAT20] and [SDI20] developed each an algorithm for fair active learning that sample data points to be labeled considering a balance between model accuracy and fairness. [BCAR+21] studies whether models trained with uncertainty-based (deep) active learning is fairer in their decisions with respect to a sensitive feature (e.g. gender) than those trained with passive learning. It appears that, with neural network architecture, active learning can preemptively mitigate fairness issues. The objective of this paper is to give a review and comparison of active learning algorithms for data labeling together with its fairness analysis on real actuarial datasets. For that purpose, this paper is focused on two areas of study: (1) address the fairness issue in Artificial Intelligence and (2) enhance the data quality in labeling datasets with active learning for insurance companies.

Section 2.2 provides a quick overview of AI governance and the fairness issue, specifically in insurance. Section 2.3 formally and mathematically defines both fairness and labeling problems. Section 2.4 gives an overview of classical active learning methods and numerically illustrates some of them on a simple synthetic dataset (cf. Section 2.4.7). In Section 2.5 we empirically study the impact of active learning methods together with a novel filtered fairness algorithm on real datasets (both fair and unfair datasets).

2.2 Fairness issue in Artificial Intelligence

Following the increase of available data (democratization of the datalakes to stock the data) and of the computing power, Artificial Intelligence (AI) constitutes a well established motivating force for the development and the transformation of the insurance sector. Indeed, the insurance use cases integrating machine learning are numerous, while the competition with new actors (GAFAM, Assurtechs) creates pressure on margins and a risk of adverse selection. Therefore the actuary must seize these new and efficient methodologies to keep and reinforce their expertise of the risk. The precision of machine learning algorithm to provide a better segmentation of risk, to achieve large scale automation (e.g. using IoT technologies, extracting information from business records), or to design a decision-making process (e.g. automation of document processing with NLP and computer vision, development of voice bot for call centers with Speech-to-Text and NLP) can both improve the risk assessment and the operational efficiency, and reduce the costs of a company. Therefore a race to deploy these AI systems has progressively taken place in insurance companies.

However, with this ever-evolving AI technology and application, one major drawback is the lack of interpretability. This may explain why adoption of AI into actuarial sciences has been slower than in other fields. These black-box algorithms make it difficult to ensure that the model does not discriminate nor induce unintentional bias, that may expose the company to operational and reputational risks.

Notably for EU's insurers, the fairness principle is an obligation. Indeed, it is recognised in the *Insurance Distribution Directive* (IDD) where it states that insurance distributors shall "always act

honestly, fairly and professionally in accordance with best interests of their customers" (Art. 17(1) - IDD). The fairness principle is also stated in Article 5 of *General Data Protection Regulation* (GDPR): personal data shall be "processed lawfully, fairly and in transparent manner in relation to the data subject" (Art. 5(1) - GDPR). For example, since the decision of the European Court of Justice of December 21, 2012, EU's insurers must no longer use gender criteria in in the computation of insurance premiums.

Thus the development of AI governance framework becomes essential to mitigate (or, if possible, to remove) some ethical issues. The idea behind such governance is to render the adoption of AI systems fair in terms of ethics in technological advancement. One of the biggest challenge in establishing governance is the Machine Learning (ML) model fairness assessment. AI systems learn to make decisions based on labeled data, which can include biased human decisions or reflect social inequities even if the sensitive variables (like gender or age) are removed. For instance, Amazon's recruitment algorithm powered by AI may implicitly favoured male applicants[4] (*gender bias*). It was also the case in 2019 of the Apple Card application developed by Goldman Sachs that could offer higher credits in men than in women with the same credit history and fiscal conditions. AI algorithms may also induce *racial bias*, such as the Microsoft's gender-recognition recognizing accurately only the gender of white men. Another example of racial bias is the software COMPAS used in courts to assess the probability that a defendant recommits a crime: a person of color will systematically be assigned a higher risk than a white person.

We note that the unfairness (e.g. gender and race) lies in the data: either the data reflect biased human decisions (e.g. the case of recruitment algorithm of Amazon) or the data is simply imbalanced leading to important errors for minority classes (e.g. the case of gender-recognition of Microsoft). Furthermore the unfairness issue becomes more significant where specific judgements of a black-blox algorithm cannot be specifically explained in a meaningful way (*transparency issue*).

Unfairness reduction algorithms can be categorized into *(1) pre-processing* in enforcing fairness directly in the data, *(2) in-processing* which enforces fairness in the training step of the learning model and *(3) post-processing* which reduces unfairness in the model inferences. As we will see in the section 2.4, a key component of *active learning* is to sample from a pool of unlabelled data. It seems natural to enforce fairness during sampling strategies, as such, falling in the category *(1)*. Contrary to the interpretability for which there does not exist yet a clear metric (cf. [MCB20, Mai21]), one can define formally an unfairness criteria as in 2.3.4. This criteria will be taken into account with respect to the machine learning algorithms and will be empirically studied in Section 2.5 both with and without a post-processing unfairness reduction algorithm.

2.3 Problem formulation

Let us provide the theoretical framework and some mathematical notations. First we introduce different measures to quantity the quality of a classification and then, we present a fairness criterion.

[4] https://www.reuters.com/article/us-amazon-com-jobs-automation-insight-idUSKCN1MK08G

2.3.1 Theoretical and empirical misclassification risk

Let us denote $\mathcal{X}$ the space of instances and $\mathcal{Y}$ the space of labels (or classes). We also consider the adequate probability measures for these spaces: $\mathcal{P}$ the distribution over $\mathcal{X}$ and $\mathcal{P}_{\mathcal{X}}$ the marginal distribution of $\mathcal{P}$ over $\mathcal{X}$. We denote $\mathcal{H} : \mathcal{X} \to \mathcal{Y}$ the space of hypotheses (also called the set of predictors). For a given labeled instance $(x, y) \in \mathcal{X} \times \mathcal{Y}$ and a given predictor $h \in \mathcal{H}$, $\hat{y} = h(x)$ is the prediction of the label of x.

A prediction is evaluated by a loss function denoted $l : \mathcal{Y} \times \mathcal{Y} \to [0, +\infty[$. For instance, a classical way for a classification problem (i.e. a task restricted to a finite set of classes $\mathcal{Y}$) is to choose the *misclassification loss* defined by $l_{0-1}(y, y') = \mathbb{1}(y \neq y')$ and the *square loss* for regression tasks (i.e. our task is to predict a continuous output) defined by $l_2(y, y') = (y - y')^2$. Note that a loss function often satisfies the following properties: for $a, y \in \mathcal{Y}$, $l(y, y) = 0$ and $l(a, y)$ is an increasing function of $|a - y|$. Interested readers can consult the appendix for more examples on loss functions.

We note that the loss function evaluates the performance of a predictor on a single observation. To evaluate the predictors on a set $\mathcal{X}$ we define the following (theoretical) risk function: for any loss function l we have

$$R(h) := \mathbb{E}[l(h(x), y)] = \int_{\mathcal{X} \times \mathcal{Y}} l(h(x), y) \mathcal{P}(dx, dy) \tag{2.1}$$

In practice, the distribution $\mathcal{P}$ is often unknown, therefore it is an analytically intractable. A (natural) estimator of this risk is its *empirical risk*: If we denote $(x_i, y_i)_{i=1}^N \subset \mathcal{X} \times \mathcal{Y}$ the observations then

$$\hat{R}(h) := \frac{1}{N} \sum_{i=1}^{N} l(h(x_i), y_i) \tag{2.2}$$

If we consider the misclassification loss then we have the following misclassification risk:

Theoretical risk: it represents a probability that the predictor h predicts a different answer than the oracle

$$R(h) = \mathbb{E}[\mathbb{1}(h(x) \neq y)] = \mathbb{P}(h(x) \neq y) \tag{2.3}$$

Empirical risk: it represents the average of times the predictor misclassify on the data

$$\hat{R}(h) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}(h(x_i) \neq y_i))$$

2.3.2 Active sampling and Empirical Risk Minimization

Let $\mathcal{D}^{(train)} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^L$ be the training set and $\mathcal{D}^{(test)} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^T$ the test set where (x_i, y_i) are drawn i.i.d. according to the distribution $\mathcal{P}$. If we assume that we have access to a large pool of unlabelled dataset denoted $\mathcal{D}_{\mathcal{X}}^{(pool)} = \{x_1^{(pool)}, \ldots, x_U^{(pool)}\}$ (we will also call it pool-set) and one or several oracles to label the data then the statistical learning process consists in the following steps:

(i) **Querying step.** This step consists of labeling the "raw" data allowing to set up labeled data (for instance both training data $\mathscr{D}^{(train)}$ and test data $\mathscr{D}^{(test)}$). We consider two types of labeling: a *passive labeling* which consists in randomly querying the data and an *active labeling* which queries the data according to an importance criterion. We note that passive labeling allows generating an i.i.d. database whereas active labeling generates a database that does not check the independence condition. The test data (also called the hold-out set) should be generated with passive labeling in order to empirically replicate $\mathscr{X} \times \mathscr{Y}$.

(ii) **Training step.** Given the training set $\mathscr{D}^{(train)}$, this step consists in finding an estimator $\hat{h} \in \mathscr{H}$ such that, for any labeled point $(x^{(train)}, y^{(train)})$, $h(x^{(train)})$ is "as close as possible" to $y^{(train)}$ while avoiding its overfitting[5]. More formally, in the training step, the objective is to find out the optimal predictor $h^* \in \mathscr{H}$ i.e. a minimizer of the theoretical risk Eq. (2.2). The theoretical risk being intractable in practice, we focus on $\hat{h}$ a function that minimizes its empirical form Eq. (2.3) instead. We call this minimization the *Empirical Risk Minimization* (ERM):

$$\hat{h} = \underset{h \in \mathscr{H}}{\operatorname{argmin}} \hat{R}_{train}(h)$$

where

$$\hat{R}_{train}(h) := \frac{1}{L} \sum_{i=1}^{L} l\left(h(x_i^{(train)}), y_i^{(train)}\right)$$

After the learning phase we expect that minimizing the empirical risk is approximately equivalent to minimizing the true risk:

$$\hat{R}_{train}(\hat{h}) \approx R(h^*)$$

(iii) **Testing step.** This step studies the performance of our estimator h on a hold-out set $D^{(test)}$. Note that this step helps detecting whether the model overfit or underfit on the training set. In a more formal way, this amounts to checking that $\hat{h}$ verifies the condition:

$$\hat{R}_{train}(\hat{h}) \approx R_{test}(\hat{h})$$

where

$$\hat{R}_{test}(h) := \frac{1}{T} \sum_{i=1}^{T} l\left(h(x_i^{(test)}), y_i^{(test)}\right).$$

In this study the model performance is not only tied to the model metric but also to the model unfairness. Hence the evaluation of a model is defined by an adequate model metric and an adequate model unfairness.

[5] too much capture of random fluctuations and variations in the training data resulting to a poor generalization of the prediction of an unlearned data

2.3.3 Precision evaluation

The choice of the metric is essential for measuring the precision of the model. The well-known accuracy (e.g. empirical form of the complement of the misclassification risk) will not be used considering its misinterpretation for imbalanced datasets: a very high accuracy might not represent a good model (e.g. a constant classifier always predicting the majority class). For that purpose in binary classification we consider simultaneously two metrics:

- **precision:** the number of true positive results divided by the number of all positive results;
- **recall:** the number of true positive results divided by the number of all samples that should have been identified as positive.

As we want the same importance of these two metrics we consider their harmonic mean **F1-score** which presents a good balance between precision and recall therefore giving good results on imbalanced classification problems. For a multi-class classification task we can compute a F1-score per class in a one-vs-rest manner and then average the results. For a more precise definition, refer to Eq. (2.6). Empirically, this metric will be further studied in Section 2.5.

2.3.4 Unfairness evaluation

As mentioned in Section 2.2, the unfairness is one of the biggest challenge in insurance (and, more broadly, in many AI governance) therefore it needs to be rigorously defined alongside with its evaluation metric. Let us first define the most widely used definitions of fairness of an estimator.

We consider $\mathcal{S} = \{-1, +1\}$ the space of sensitive feature (e.g. gender or sex). We also consider $\mathcal{X}_{-S} := \mathcal{X} \setminus \mathcal{S}$ the space of instances without excluding the space of the sensitive feature, K the total number of classes and Y the true response of the task.

Definition 2.1: Equalized Odds

In Equalized Odds (a.k.a. Positive Rate Parity) (see [HPS16]), we say that a classifier $h \in \mathcal{H}$ is fair with respect to the distribution $\mathbb{P}$ on $\mathcal{X}_{-S} \times \mathcal{S} \times [K]$ if $h(X)$ and S are independent conditional on Y. For binary classification, this definition states that protected and unprotected group should have equal true positive rates and false positive rates.

A relaxation of equalized odds is possible in binary case: we can require to have non-discrimination group only within the "advantaged" outcome (e.g. university admissions or employee recruitment). This unfairness is called *Equal Opportunity*.

Definition 2.2: Equal Opportunity

In binary setting, we say that a classifier $h \in \mathcal{H}$ satisfies Equal Opportunity (a.k.a. True Positive

Parity)(see [HPS16]) with respect to the distribution $\mathbb{P}$ on $\mathscr{X}_{-S} \times \mathscr{S} \times \{0,1\}$ if

$$\mathbb{P}(h(X) = 1 | S = 1, Y = 1) = \mathbb{P}(h(X) = 1 | S = -1, Y = 1)$$

A relaxation of the above fair definitions is possible in the multi-class case: the *Demographic Parity* which require to have non-discrimination group in all (predicted) outcomes.

Definition 2.3: Demographic Parity

In Demographic Parity (a.k.a. Statistical Parity) see [CKP09], we say that a classifier $h \in \mathscr{H}$ is fair with respect to the distribution $\mathbb{P}$ on $\mathscr{X}_{-S} \times \mathscr{S} \times [K]$ if

$$\mathbb{P}(h(X) = k | S = 1) = \mathbb{P}(h(X) = k | S = -1), \quad \forall k \in [K] \ .$$

This definition states that protected and unprotected group should have equal likelihood. In an approximate version we want: for a given small $\epsilon > 0$,

$$|\mathbb{P}(h(X) = k | S = 1) - \mathbb{P}(h(X) = k | S = -1)| \leq \epsilon, \quad \forall k \in [K].$$

In this study, we consider multi-class classification problems under **Demographic Parity** fairness constraint [CKP09], that requires the independence of the prediction function from the sensitive feature S for all classes.

Evaluation in Demographic Parity. Given a classifier $h \in \mathscr{H}$, when fairness is required, two important aspects of the classifier need to be controlled: its misclassification risk R defined in Eq. (2.1) and its unfairness that will be evaluated by

$$\mathscr{U}(h) := \frac{1}{K} \sum_{k=1}^{K} |\mathbb{P}(h(X) = k | S = 1) - \mathbb{P}(h(X) = k | S = -1)| .$$

Naturally, taking into account the definition above, a classifier h is more fair as $\mathscr{U}(h)$ becomes small. This quantity being intractable, its empirical form (see Eq. (2.7)) will be used in practice.

2.4 Active Learning methods

For the present study let us introduce the active learning setting before studying its sampling methods.

2.4.1 Definitions and framework

Given a set of hypothesis $\mathscr{H}$, *active learning* consists in iteratively querying oracle to label instances $x \in \mathscr{D}_{\mathscr{X}}^{(pool)}$ that provides the most information for learning an hypothesis $h \in \mathscr{H}$. Informally, active learning enables a learning model to perform better with fewer labeled data if we can query from a

pool of unlabelled data. Following each query the model is trained and updated w.r.t. the new labelled data. There are several ways to access unlabeled data, among them there are:

1. the *offline scenario* where the raw data is directly accessible in large quantities, for example, product reviews left by Internet users on a website. The query strategy associated to this scenario is called *pool-based sampling* (or *offline sampling*). In this scenario, we assume that the unlabeled data $\mathscr{D}_{\mathscr{X}}^{(pool)} \subset \mathscr{X}$ is completely available. Given a trained model $h \in \mathscr{H}$ the goal of an active learner would be querying the most informative instance according to a well-defined importance score $I(\cdot, h)$. We call *batch-mode sampling* the procedure of querying more than one instance per active learning iteration. A natural approach for a batch-mode sampling would be querying the top instances according to the importance score.

2. the *online scenario* where the data is collected one by one, for example, e-mails received in an e-mail box. The query strategy associated to this scenario is called *stream-based sampling* (or *online sampling*). In this scenario, the active learner decides whether each new instance $x^{(stream)}$ should be queried or not. For each accepted query, we obtain the label $y^{(stream)}$ of $x^{(stream)}$ (via the oracle). Given an importance score function, a natural approach for online sampling would be querying instances that have their score above a given threshold.

The goal of active learning in both scenarios is to retain the optimal set of queries that maximizes the model's performance (e.g. minimizing the empirical risk on test-set). In contrast, the traditional model of supervised learning is trained on a dataset queried randomly (passive sampling) from the pool-set. This last process is called *passive learning*.

In active learning, the objective is to build actively a model more accurate than the passive one. In other words, if we denote $\hat{h}_a, \hat{h}_p \in \mathscr{H}$ the models trained respectively by active and passive learning process with the same cardinal of training set, then we expect to have

$$\hat{R}_{test}(\hat{h}_a) < \hat{R}_{test}(\hat{h}_p).$$

About the training step. Let $h_t \in \mathscr{H}$ be the previous trained model. We assume that (x^*, y^*) is the current queried data. During the training step of the active learner, we can either (1) construct h_{t+1} in re-training h_t on the whole labeled set including (x^*, y^*) (called *batch learning* approach) or (2) construct h_{t+1} in updating the weight of h_t based only on (x^*, y^*) (called *online learning* approach. See [SS+11] for more details).

About the learning task. In the current literature, each query strategy often depends on the learning task we are studying: sampling strategies for a classification problem often differ from those for a regression. This is mainly due to the nature of the model responses: unlike regression, a probabilistic classification model often gives a directly interpretable response (for instance a posterior probability

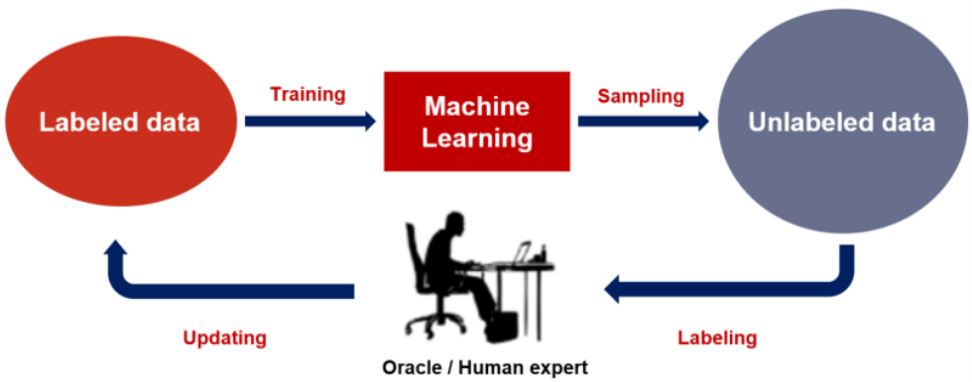


Figure 2.1: Active learning in an offline scenario.

that a given class is correct) in terms of uncertainty therefore leading to some natural heuristic choices for the importance score function.

In practice, although some tabular data require the intervention of an additional tool for labeling, active learning sees its usefulness in the labeling of unstructured data such as texts, images or sounds. Indeed unstructured data often suffer from a lack of labeling due to its difficulty and its high complexity in time and cost. Labeling unstructured data involves the intervention of oracles / human experts (hence the *cost*) that may require some expertise[6] (hence the *difficulty*) to label the data one by one (hence the *time*), especially when the data is in sound, video or text format.

Settings. Motivated by the above practical insights we study the active learning for (both binary and multi-class) **classification tasks**. Furthermore, we assume that unlabeled data is inexpensive and abundant but labeling them is difficult, expensive and time consuming. Therefore we are in an **offline scenario**. Given an importance score function I, we present in Algorithm 2 the active learning process.

[6]For instance some textual data in insurance need to be studied by legal experts, e.g. the GDPR-compliancy.

Algorithm 2 Outline of an active learner process in an offline scenario

Input: h a base estimator, $\mathscr{D}^{(train)}$ the initial training set and $\mathscr{D}_{\mathscr{X}}^{(pool)}$ the initial pool-set.

Step 0. Fit h on the training set $\mathscr{D}^{(train)}$.

Step 1. The active learner query the instance $x^* \in \mathscr{D}_{\mathscr{X}}^{(pool)}$ that maximizes the importance score function $I(x,h)$ i.e.

$$x^* = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \{I(x,h)\}$$

Step 2. Update the training set and the pool-set: if we denote y^* its label then

$$\begin{aligned} \mathscr{D}^{(train)} &= \mathscr{D}^{(train)} \cup \{(x^*, y^*)\} \\ \mathscr{D}_{\mathscr{X}}^{(pool)} &= \mathscr{D}_{\mathscr{X}}^{(pool)} - \{x^*\} \end{aligned}$$

Step 3. As long as we do not reach a stopping criterion (e.g. exhaustion of the labeling budget or convergence of the performance), we repeat this process (**return to step 0**).

Output: the final estimator h

Some statistical guarantees. Let us denote $h^* \in \mathscr{H}$ the Bayes classifier with d its Vapnik-Chervonenkis (VC) dimension (see [VC15]), $\hat{h}_a \in \mathscr{H}$ the active learner and $\hat{h}_p \in \mathscr{H}$ the passive learner. The VC dimension is a measure of a learning algorithm's "complexity": it is defined by the cardinality of the largest set of points that the algorithm can shatter (given any labeled data points the algorithm can always learn a perfect classifier). We also denote classification error excesses:

$$\varepsilon_p = R(\hat{h}_p) - R(h^*) \quad \text{and} \quad \varepsilon_a = R(\hat{h}_a) - R(h^*).$$

Then, it is shown by [BBL09] that under good conditions on the label distribution like the *Massart noise*[7] we have the following convergence rates (n being the number of labeled data) :

$$\varepsilon_p \sim \frac{d}{n} \quad \text{and} \quad \varepsilon_a \sim \exp\left(-\text{constant} \times \frac{d}{n}\right). \tag{2.4}$$

This means that (under this condition) active learning outperforms passive learning and therefore reduces the labeling costs. Note that these results are achieved by a disagreement-based active learning algorithm called A^2 (*Agnostic active learning*) algorithm, as detailed in [BBL09]. The notion of disagreement-based sampling will be defined and studied in Section 2.4.3.

In the next sections we will introduce the various classical approaches in active learning to define I before studying them on both synthetic and real datasets.

[7] The *Massart noise* verifies: $\exists \beta < 1/2$ such that $\mathbb{P}(Y \neq h^*(X)|X) \leq \beta$ almost everywhere with $X \in \mathscr{X}$ and $Y \in \mathscr{Y}$.

2.4.2 Sampling based on uncertainty

This approach consider that the most informative instances are the ones located in the region of uncertainty of the learning model. Therefore, the idea is to query and learn the most uncertain sample according to the estimator. Intuitively this approach tends to avoid querying redundant instances since the model learned on an uncertain data point will probably be more certain. Let us formally define the uncertainty of an instance.

Given the current predictor $h_t \in \mathscr{H}$ (t being the $t-th$ update of the model) and an instance $x \in \mathscr{X}$, we denote $p_{t,x}(y) := \mathbb{P}(h_t(x) = y|x)$ the probabilistic response of classifying x as y according to h_t. This approach is studied under both binary and multi-class classification task.

Binary classification. In a binary task, a natural method is to sample the least confident instances (LC) for our learning model. In a more formal way we query the instance

$$\hat{x}_{\text{LC}} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmin}} \left\{ \left| p_{t,x}(y) - \frac{1}{2} \right| \right\}$$

where the *un*importance criterion is defined by

$$I_{\text{LC}}(x, h_t) = \left| p_{t,x}(y) - \frac{1}{2} \right|$$

Thus defined, the queried instance $\hat{x}_{\text{LC}}$ is in the region of uncertainty of h_t. We therefore hope, via this approach, that the updated model h_{t+1} will either perform better (section 2.3) than the unupdated model h_t or perform better than passive learning.

Multi-class classification. Let $K > 2$ be the number of classes in a multi-class classification setting. A naive approach would be to generalize the approach presented in the binary case by querying the most uncertain instance according to

$$\hat{x}_{LC} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmin}} \left\{ p_{t,x}(y^{(K)}) \right\}$$

Where we denote $y^{(k)}$ the argument of the k-th highest value of $p_{t,x}$. It follows $p_{t,x}(y^{(1)}) \leq p_{t,x}(y^{(2)}) \leq \cdots \leq p_{t,x}(y^{(K)})$ and therefore $y^{(K)}$ is highest probable class for the instance x under the model h_t,

$$y^{(K)} = \underset{y \in [K]}{\operatorname{argmax}} \; p_{t,x}(y)$$

$\hat{x}_{LC}$ thus is the queried instance with the smallest probability (in the pool-set) of the most probable class. Note this approach focus only on $y^{(K)}$ and ignores the distribution of the other classes $y^{(k)}$, $k \neq K$ and therefore it raises the following issue: for a given instance x if the margin between $p_{t,x}(y^{(K)})$ and $p_{t,x}(y^{(K-1)})$ is small then x can be considered as an *uncertain* data point *undetected* by the the LC criteria. In contrast, a given instance can be, naturally, considered as a *certain* data point if there is a

large margin between class probabilities. The Shannon entropy (defined in [Sha48]) can overcome this problem.

The Shannon entropy is often used in statistical learning as a measure of uncertainty (as for example in the calculation of decision trees with the entropy measure). In this case, the Shannon entropy assigns to a given instance x an uncertainty score according to the distribution of $p_{t,x} = \big(p_{t,x}(1), p_{t,x}(2), \cdots, p_{t,x}(K)\big)$. This measure classifies instances according to whether the posterior probability distribution of the labels is uniform. More precisely a distribution close to the uniform distribution underlines the difficulty for the model to decide on the right class while a distribution far from the uniform distribution implies a more reliable prediction. For any instance x and fixed time t we note the uncertainty score $H_{t,x} := H_{t,x}(Y)$ with Y the discrete random variable on $\mathscr{Y}$ where each event $\{Y = y\}$ occurs with probability $p_{t,x}(y)$.

The entropy-based uncertainty score is defined by $H_{t,x} = -\sum_{k=1}^{K} p_{t,x}(k) \log p_{t,x}(k)$ and we thus query the instance that verifies the equation

$$\hat{x}_H = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ H_{t,x} \right\}.$$

The corresponding sampling is called uncertainty sampling. In the definition of $H_{t,x}$ we use the convention $0 \log 0 = 0$.

Let us denote $I_Y(y) = -\log p_{t,x}(y)$ the information (or *self-information*) that the random event $\{Y = y\}$ contains. We remark that more uncertain an event is, the more informative it is (thus interesting) and a certain event contains no information. We note that the entropy can also be written as

$$H_{t,x}(Y) = \mathbb{E}[-\log p_{t,x}(Y)] = \mathbb{E}[I_Y].$$

The Shannon entropy of a random variable Y is therefore the expectation of the information contained in the variable Y.

Although these uncertainty-based methods can strengthen the learning model, especially on regions of uncertainty, they remain difficult to use when the model is not enough reliable for prediction. Indeed, the current training set may not contain sufficient information for the model: we call this problem, the *cold-start problem*[8]. Moreover, this sampling method works only for probabilistic model and therefore is not adequate for deterministic or non-probabilistic models (e.g. SVM). Alternative methods exist to overcome (at least partially) these problems such as disagreement-based sampling.

2.4.3 Sampling based on disagreement

For an active learning process the choice of the initial labelled data is crucial so as to get an accurate uncertainty score. If the dataset is insufficiently representative of $\mathscr{X}$ then too many plausible model parameters can be suggested for fitting such dataset leading therefore to a *model robustness* issue. We

[8]The term comes from the analogy with starting very cold engines such as cars. Cold starting a car can be difficult to deal with but as soon as the engine is running the car starts to move forward and the performance increases.

call this phenomenon a *high epistemic uncertainty*. By contrast a *low epistemic uncertainty* indicates a robust model. Usually epistemic uncertainty arises in areas where there are fewer samples for training.

Instead of relying on the uncertainty measure based on a single model, disagreement-based sampling proposes a more robust method by combining the result of several learning models all different from each other (so-called *ensembling* methods). The idea described in [SOS92] is to rely on a set of models to "vote" the informativeness of each instances. This set of models is named a *Committee*. For a committee an informative instance is characterized by the highest voting disagreement among the models. For a given instance, a model's vote can be characterized either by its label prediction (a.k.a. *"hard" vote*) or by its label posterior probability (a.k.a. *"soft" vote*). This approach is called *Query-By-Committee* (a.k.a. *QBC*). QBC requires two main components, namely the construction of the **committee** and the definition of the **disagreement measure** answering respectively the following questions: (1) how can we define a set of models close enough to sample the adapted regions of uncertainty but different enough to ensure the informativeness of the queries ? (2) How to measure the degree of disagreement of the Committee ?

Committee. Consider a committee of models $h_t^{\text{committee}} = \{h_t^1, \ldots, h_t^C\}$ where each $h_t^i \in \mathscr{H}$ is trained on the current training-set $\mathscr{D}^{(train)}$. QBC is a query strategy based on the maximum disagreement of $h_t^{\text{committee}}$ constructed by randomized copies of a learned model. At each iteration, the algorithm generates a new committee of classifiers based on the updated training-set.

Initially in his simulations [SOS92] had implemented a Gibbs training to learn two perceptrons: each perceptron is consistent with the training-set $\mathscr{D}^{(train)}$ with slightly different parameters due to the randomness of the Gibbs algorithm. These generated models query the instance where their predictions are the most dispersed. Note that [FST97] showed that for a given performance (see Eq. (2.4)) and given some assumptions this process leads to an exponential decrease in the number of labeled instances required compared to passive learning. However, in spite of these theoretical guarantees, Gibbs algorithms has a high computational complexity and is hard to implement for more complex learning models. We overcome this problem by using other approaches to generate $h_{\text{committee}}$:

- For generative models such as linear discriminant analysis (LDA) or naive Bayesian classification, we can randomly sample an arbitrary number of models from a $h \rightarrow \mathbb{P}(h|\mathscr{D}^{(train)})$ distribution with $h \in \mathscr{H}$. As an example, [DE95] samples Hidden Markov Models (HMM) using normal distribution;

- For other classes of models, such as discriminative (e.g. logistic regression) or non-probabilistic (e.g. SVM) models, Ensemble methods such as Bagging or Boosting are used for constructing the committee. These query strategies were first proposed by [AM98] and are defined as follows:

  Query-by-bagging. Let $h \in \mathscr{H}$ be a base estimator and $\mathscr{D}^{(train)}$ our current training-set. The first step of *query-by-bagging* method consists in bootstrapping $\mathscr{D}^{(train)}$ into C sets of the same size as the original set denoted by $\mathscr{D}^{(train),1}, \ldots, \mathscr{D}^{(train),C}$. Then we construct the committee

of models $h^{\text{committee}} = \{h^1, \dots, h^C\}$ such that each member h^i corresponds to the base model h trained on the bootstrapped training-set $\mathscr{D}^{(train),i}$.

Query-by-boosting. Let $h \in \mathscr{H}$ be a base estimator and $\mathscr{D}^{(train)}$ our current training-set. The method *query-by-boosting* consists in building the committee $h^{\text{committee}} = \{h^1, \dots, h^C\}$ via *Adaboost* (*Adaptive Boosting* [FS97]) on the base model h where $(h^1, \dots, h^C)$ is a sequence of copies of h trained on an iteratively modified dataset: the weights of incorrectly classified instances are adjusted such that following classifiers focus more on challenging examples. Query-by-boosting approach chooses an instance for which the weighted vote obtained by boosting is the most dispersed.

Disagreement measure. Given a committee $h_t^{\text{committee}} = \{h_t^1, \dots, h_t^C\}$ with t the number of updates (or active learning iterations), there exists various disagreement measures for evaluating the dispersion of votes for multi-class classification problems. In the active learning litterature we have the following known methods:

- *vote by entropy* proposed by [DE95] is an entropy-based method combined with a *hard* committee vote:

$$\hat{x}_{VE} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ -\sum_{k=1}^{K} \frac{v_{t,x}^{\text{committee}}(k)}{C} \log \frac{v_{t,x}^{\text{committee}}(k)}{C} \right\}$$

  where $v_{t,x}^{\text{committee}}(k) := \sum_{c=1}^{C} \mathbb{1}\{h_t^c(x) = k\}$ is the number of *hard* votes of the committee for the label k given the instance x.

- the *mean Kullback-Leibler (KL) divergence* [MN98], is a method based on the KL divergence combined with a *weak* committee vote:

$$\hat{x}_{KL} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ \frac{1}{C} \sum_{c=1}^{C} D(\, p_{t,x}^{c} \,||\, p_{t,x}^{committee} \,) \right\}$$

  where for all $c \in [C]$, $D(\, p_{t,x}^{c} \,||\, p_{t,x}^{committee} \,)$ is the KL divergence defined by

$$D(\, p_{t,x}^{c} \,||\, p_{t,x}^{committee} \,) = \sum_{k=1}^{K} p_{t,x}^{c}(k) \log \left\{ \frac{p_{t,x}^{c}(k)}{p_{t,x}^{committee}(k)} \right\}$$

  with $p_{t,x}^{c}(k) = \mathbb{P}(h_t^c(x) = k|x)$ the probability that the model h_t^c predicts that the class of x is k and $p_{t,x}^{committee}(k) = \frac{1}{C} \sum_{c=1}^{C} \mathbb{P}(h_t^c(x) = k|x) = \frac{1}{C} \sum_{c=1}^{C} p_{t,x}^{c}(k)$ is an averaged (over all committee member) probability that k is the correct class.

So far, we have presented techniques for querying instances that lie on an uncertain regions of classification. These uncertain areas are characterized either by the uncertainty of a classifier (iterative update of a *single set* of parameters) or by the disagreement of a model committee on its classification (iterative update of *several sets* of parameters). However, these approaches intend to improve only a local (and not a global) prediction quality of an estimator. Indeed, these methods do not take into account the influence of a queried input instance $\hat{x}$ on the other parts of the input space: implicitly they try to increase the quality of the model near $\hat{x}$. On the other hand the following algorithms propose to query the instances that "impact" the most learning models.

2.4.4 Sampling based on model change

The idea of these approaches is to choose the instance that gives the most change (or impact) on our learning model if we know its label. The word "impact" may seem vague but it highlights a class of possibilities in terms of sampling criteria. In the current literature, when we know its label a candidate instance can impact the model mainly in two manners: (1) *impact on the model parameters* and (2) *impact on the model performance.*

2.4.4.1 Impact on the model parameters

The idea is to query instances that can change the learning model $h_t \in \mathscr{H}$ as much as possible. This can be done by evaluating the change of the model parameters between the updated model h_{t+1} and the current model h_t. Intuitively, if an instance is able to modify considerably the parameters of a model, then this instance contains information on the underlying distribution $\mathscr{X}$ which is not (or rarely) found in the training-set. In the following, we call this set of strategies the *Expected Model Change* (EMC). An example of a change measure is the *Expected Gradient Length* (EGL).

Expected Gradient Length. The EGL strategy applies to all learning models that require the computation of the gradient of a loss function during training (e.g. training via gradient descent). Consider l_t a loss function with respect to the model $h_t \in \mathscr{H}$ and ∇l_t its gradient. The degree of change of the model can be measured by the Euclidean norm of the training gradient $\|\nabla l_t(\mathscr{D}^{(train)})\|$: if we denote $\mathscr{D}_t^{(train)}$ and $\mathscr{D}_{t+1}^{(train)}$ the data labeled at iteration t and $t+1$ respectively then a "small" impact results in a norm $\|\nabla l_t(\mathscr{D}_{t+1}^{(train)})\| \approx \|\nabla l_t(\mathscr{D}_t^{(train)})\| \approx 0$ while a "large" impact results in a large margin between these two norms. Thus, the instance to query is the instance x which, if labeled and added to $\mathscr{D}^{(train)}$, results (on average over the set of possible labels) in a larger gradient size. Formally this amounts to querying x which maximizes the term

$$\mathbb{E}\left[\|\nabla l_t\left(\mathscr{D}^{(train)} \cup (x, Y)\right)\|\right]$$

with $Y \sim p_{t,x}$ a discrete random variable on $\mathscr{Y}$. That is

$$\hat{x}_{EGL} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} p_{t,x}(k) \| \nabla l_t \left(\mathscr{D}^{(train)} \cup (x,k) \right) \| \right\}.$$

For the sake of time complexity we approximate $\hat{x}_{EGL}$ by

$$\tilde{x}_{EGL} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmax}} \left\{ \sum_{k=1}^{K} p_{t,x}(k) \| \nabla l_t ((x,k)) \| \right\}.$$

Indeed after training the model h_t on $\mathscr{D}^{(train)}$, $||\nabla l_t(\mathscr{D}^{(train)})|| \approx 0$ (l_t reaches a local minimum) and we often assume that the training-set is independent.

2.4.4.2 Impact on the model performance

The idea is to query instances that can reduce the generalization error. We can reduce the forecast error by estimating this error directly empirically (e.g. *Expected Error Reduction*) or indirectly by reducing the variance present in the risk of a learning model.

Expected Error Reduction. The Expected Error Reduction (EER), proposed by [RM01], is a strategy consists in choosing the instance that minimizes the *expected* of generalization error since the class of the instance is currently unknown. Let us note

- h_t a predictor of $\mathscr{H}$ trained on $\mathscr{D}_t^{(train)}$ at time t;
- $h_{t+1}^{(x,k)}$ the updated predictor re-trained on $\mathscr{D}^{(train)} \cup (x,k)$ at time $t+1$;
- $p_{t+1,u}^{(x,k)}(v) = \mathbb{P}(h_{t+1}^{(x,k)}(u) = v|u)$ the probability that the class of u is v under $h_{t+1}^{(x,k)}$;
- $y^{(K)} = \underset{y \in \mathscr{Y}}{\operatorname{argmax}} \; p_{t+1,u}^{(x,k)}(y)$.

[RM01] proposed the following approach for minimizing the expected error (based on 0-1 loss):

$$\hat{x}_{EER} = \underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\operatorname{argmin}} \left\{ \sum_{u \in \mathscr{D}_{\mathscr{X}}^{(pool)}} \mathbb{E}_Y \left[1 - p_{t+1,u}^{(x,Y)}(y^{(K)}) \right] \right\}.$$

The main drawback of this method is its time complexity. Indeed this sampling method involves $\left| \mathscr{D}_{\mathscr{X}}^{(pool)} \right| \times K$ re-training of h_t.

Thus far we have presented querying techniques that are based either on the impact of the instances on the learning model, either on the overall predictive quality of the model or on the degree of expected model change. In the next section we study a complementary sampling strategy based on the representativeness of the instances.

2.4.5 Sampling based on representativeness

All the sampling methods presented so far aim at choosing the instance that gives the highest informativeness to the learning model: an informativeness based on the quality of the local prediction (cf. Section 2.4.2 and 2.4.3) or global prediction (cf. Section 2.4.4). However, some of these instances may not be representative of the distribution of $\mathscr{X}$ leading to a possible decrease of the model performance. As an example, it is possible that in our pool-set $\mathscr{D}_{\mathscr{X}}^{(pool)}$ we have anomalies such as *outliers* which are not representative of $\mathscr{X}$ distribution but which are possibly considered informative in the sense of the sampling approaches presented above[9]. Thus, this subsection introduces methods allowing to take into account the representativeness of the queried data: an informativeness component alone is not enough and the approaches presented here propose to add a representativeness component (an instance must thus verify a good compromise between informativeness and representativeness). An example of this methodology, called *Information Density*, is presented by [SC08a] and formally defined in Eq. (2.5).

We denote $I_A : \mathscr{X} \times \mathscr{H} \to \mathbb{R}$ an importance criterion (for the query) according to an informativeness measure A (e.g. entropy sampling). Let I_R be a representativeness (or density) measure. Let us choose the following measure: for all $x \in \mathscr{X}$

$$I_R(x) = \frac{1}{|\mathscr{D}_{\mathscr{X}}^{(pool)}|} \sum_{u \in \mathscr{D}_{\mathscr{X}}^{(pool)}} \text{sim}(x,u) \tag{2.5}$$

with sim a similarity measure between two instances (e.g. the cosine similarity). $I_R(x)$ measures the average similarity between the instance x and the instances of the set $\mathscr{D}_{\mathscr{X}}^{(pool)}$.

Information Density. A sampling strategy that relies on this I_R measure is the density weighted methods proposed by [SC08a]:

$$\hat{x}_{ID} = \left(\underset{x \in \mathscr{D}_{\mathscr{X}}^{(pool)}}{\text{argmax}}\, I_A(x,h) \right) \cdot I_R(x)^{\beta}$$

with β a parameter tuning the importance given to the representativeness measure. Taking into account both informativeness and representativeness, the instances thus selected have a low predictive quality and are much requested.

There are many other active learning methods that take into account the representativeness and the density of the instances. [SC08a] proposed a variant of the weighted density method by integrating a clustering method: first they cluster the set $\mathscr{D}_{\mathscr{X}}^{(pool)}$ before computing the average similarity with the instances of the same cluster.

In this section, we have cited the most common query strategies in the current active learning literature. In the next subsection, we present some recent active learning methods based on deep learning models.

[9] An outlier may be in the area of model uncertainty or may result in a significant impact on the model parameters after it is labeled.

2.4.6 Sampling based on neural nets architecture

In a "classical" active learning setup, Deep Learning methods raise additional challenges:

1. Active learning methods rely on training models on a small amount of labeled data whereas recent Deep Learning algorithms are increasingly greedy in terms of labeled data due to the explosion of the number of parameters. Therefore a too complex neural network architecture can overfit to a "simple" data. Moreover, the *cold-start* problem stated in section 2.4.2 may intensify this issue and consequently may render the previous sampling methods unusable.

2. Most active queries are based on the uncertainty given by the model. However, standard Deep Learning algorithms for classification (or regression) problems do not capture well uncertainties. Indeed, in multi-class classification, the probabilities obtained with a *softmax* output layer (which is often the case in recent architectures) are often misinterpreted as the confidence in the model. [GG16] shows that the *softmax* function results in extrapolations with unjustifiably high confidence for points far from the training data.

Thus, this section presents existing approaches in active learning literature to overcome these difficulties.

Contrary to the previous sections, it is natural to present approaches that are not agnostic to learning models. Indeed the following methods are designed for neural network architectures and can be divided into two categories: *uncertainty-based* sampling and *representativeness-based* sampling approaches.

Uncertainty of instances. We recall that our goal is to estimate an uncertainty score for each unlabeled instance (see section 2.4.2 and 2.4.3). However, as mentioned above neural networks do not capture uncertainty (or only to a limited extent). Nevertheless, recent studies have shown that uncertainties can be estimated by the introduction of Bayesian methods in neural networks. We call them *Bayesian Deep Learning* methods. Applying Bayesian methods on neural networks to approximate as closely as possible the true distribution of the posterior probability of the parameters $\mathbb{P}(w|\mathcal{D}^{(train)})$ (with w the parameters of the neural networks) have been extensively studied in the literature. Recently, [GG15b] proposes to rely on recent advances in *Bayesian Deep Learning* for the estimation of the uncertainty of the learning model. The principle is as follows: if the probabilistic model of neural networks is defined by $h(x,w)$ with $x \in \mathcal{X}$ and w the parameters (or weights) then $\mathbb{P}(w)$ is the a priori distribution on the space of the parameters (often the a priori law is Gaussian $\mathbb{P}(w) = \mathcal{N}(w|\mu,\sigma^2)$) and the likelihood is often defined by a *softmax* $\mathbb{P}(y = c|x,w) = \text{softmax}(h(x,w))$. Our goal is therefore to find (or approximate) the following posterior distribution on w:

$$\mathbb{P}(w|X,Y) = \frac{\mathbb{P}(Y|X,w)\cdot\mathbb{P}(w)}{\mathbb{P}(Y|X)}$$

[GG15a] proposes to approximate this value by variational inference. Later, [GG16] proposed to use the regularizer *dropout* in deep networks as a Bayesian approximation of a Gaussian process. In

their paper, they showed that any model trained with *dropout* is an approximation of a Bayesian model, and the estimated uncertainty is the variance of multiple predictions (output of the *softmax* layer) with different *dropout* filters: this method is also known as *Monte Carlo Dropout* (MC-Dropout). We note that for this methodology the dropout regularizer is used both in the learning phase ("classical regularizer") and in the inference phase (the predictions are therefore no longer deterministic). [LPB17] on the other hand proposes a non-Bayesian methodology by considering a *ensembling* approach to estimate this uncertainty. Specifically, the authors of the paper propose to train a set of the same neural network architecture (but with different randomly initialized initialization weights) and take the average of the *softmax* vectors.

Representativeness of instances. The objective is to estimate a representativeness score of the instances: the selected instances must fairly represent the underlying distribution $\mathscr{X}$. [SS18] proposes density-based methods. To this end, the authors propose to define the active learning problem as a selection *core-set*: finding a small subset of a large labeled dataset such that a model learned on the small dataset is competitive on the entire labeled dataset. The idea is to choose c centroids such that the largest distance between these crentroids and the rest of the unlabeled data is as small as possible. However, the *core-set* approaches require to compute a large distance matrix on the unlabeled data which leads to a high computational complexity.

2.4.7 Numerical illustrations

In this section we discuss several numerical aspects of the studied procedure on synthetic datasets. passive learning is used as a benchmark. In a nutshell we illustrate the efficiency and behaviour of different active learning processes to build a ML model.

Toy examples. All studies in this section are realized on Gaussian datasets called *Two Gaussians*. We have generated 2000 Gaussian instances of dimension 2: 1000 *red* examples and 1000 *blue* examples with same variance but different means. These examples will then constitute the set of labeled data (red or blue points) and unlabeled data (gray points). At initialization (iteration 0 of AL) 10 instances are randomly queried for constructing the first training set (see Figure 2.2). Note that since the dataset is balanced the metric accuracy (*acc*) is used for comparing the models performances.

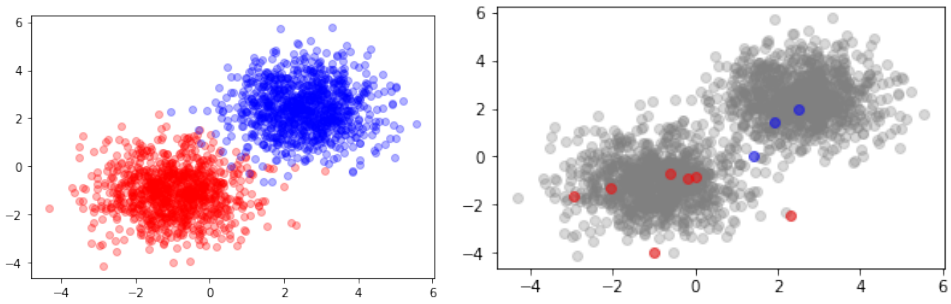


Figure 2.2: Synthetic dataset: Two Gaussian.
Left: fully labeled dataset. Right: pool-set (gray points) and train-set (red or blue points).

Results. Figure 2.3 displays the behaviour of passive learning on the synthetic dataset. This figure is used as a benchmark. After 3 iterations, passive learning updates slowly the estimator keeping roughly the same accuracy whereas Figures 2.4, 2.5, 2.6 and 2.7 illustrate how active learning outperforms the classical querying way. Uncertainty and disagreement based sampling (Figures 2.4 and 2.5) focus on the model uncertainty to classify. The loss-based sampling (Figure 2.6) focus more on the *non-dense* tails of the two normal distributions (near model uncertainty regions but far from dense regions). At last, the density-based sampling (Figure 2.7) query both uncertain and representative (near dense regions) instances.

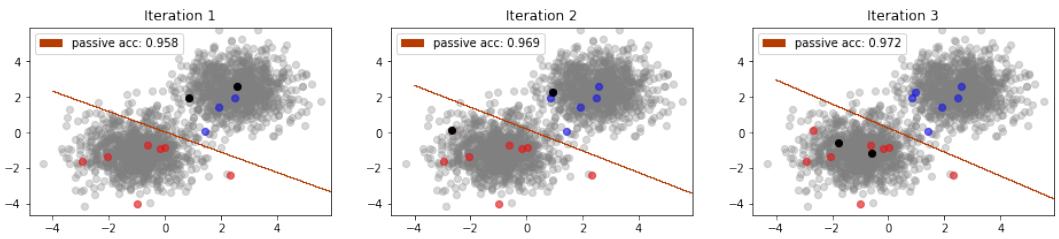


Figure 2.3: passive learning on synthetic data.
Here we illustrate the first 3 iterations of passive learning with, for each iteration, a random querying of two instances (in black). The green line illustrates the decision threshold of the logistic model after training on the labeled data.

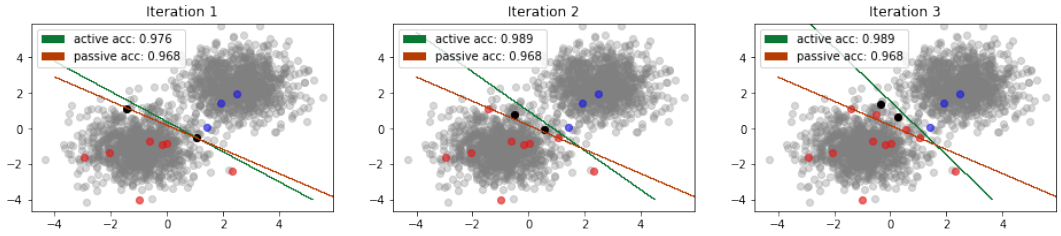


Figure 2.4: Sampling by Shannon Entropy.
We illustrate here the first 3 iterations of active learning with, for each iteration, two instances (in black) queried according to their uncertainty entropy-based score. The *green line* illustrates the decision threshold of the logistic model after training on the labeled data and the *red line* the last updated passive model in Figure 2.3.

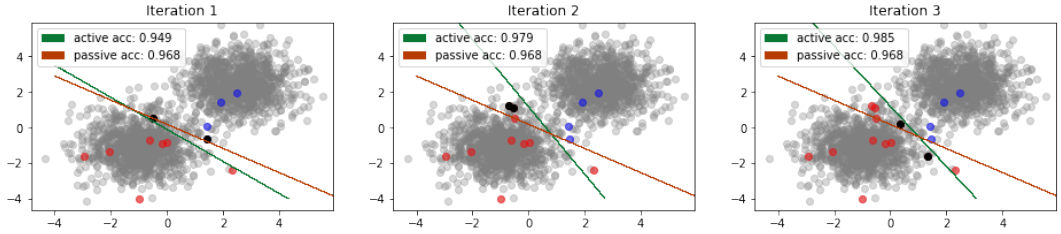


Figure 2.5: Sampling by Query by Bagging.
We illustrate here the first 3 iterations of active learning with, for each iteration, two instances queried according to their disagreement score. Here, the disagreement measure used is the **vote entropy** and the model committee was built by the **query by bagging** method by replicating 8 logistic regression models.

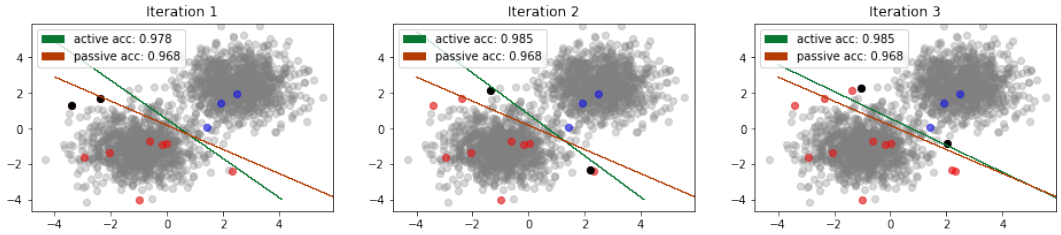


Figure 2.6: Sampling by Expected Gradient Length.
We illustrate here the first 3 iterations of active learning with, for each iteration, two instances queried according to their Expected Gradient Length score used on a binary logistic model.

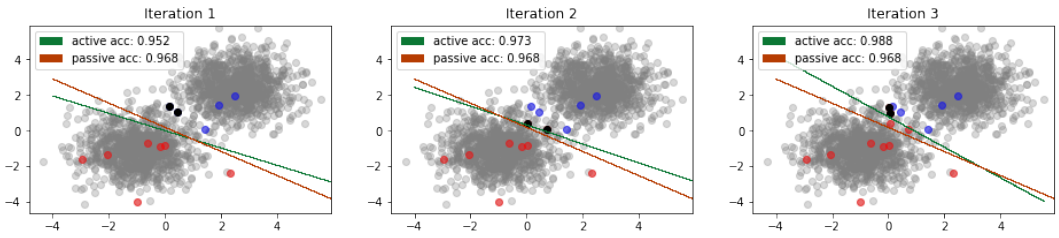


Figure 2.7: Sampling by Information Density.
We illustrate here the first 3 iterations of active learning with, for each iteration, two instances queried according to their representativeness (and uncertainty) score. The representativeness measure used is the Information Density on a logistic model.

2.5 Application on real datasets

In this section, we will present the empirical effectiveness of active learning methods on real data. In particular, we will apply them on textual data and evaluate their performance.

2.5.1 Practical considerations

For several years now, research in active learning has been progressing but the proposed methodologies remain a challenge in terms of feasibility in practice. Indeed, experimental active learning is often difficult for researchers because they do not have access to a labeling oracle. Thus, in order to evaluate the effectiveness of the proposed methodologies, a widely used trick is to take a labeled database and transform it into a database suitable for active learning methods by "masking" the label of the data that is not queried by active querying. To build up unlabeled data, we need to "hide" their label. This naturally raises to some important issues in practice that must be considered when implementing active learning methodologies:

- *How to start labelling?* The problem of *cold-start* remains a challenge: we want to have enough representative data for our learning model before starting active querying. Indeed, if the learning model is poorly calibrated, the selected instances may be less informative than a passive query. For the initialization step, [ST11] and [RS13] propose for example to query representative instances first in order to have a large decision scope and uncertainty-based queries are then used to frame the decision scope and improve learning performance.

- *How to evaluate active learning methodologies?* In practice it is difficult to compare them when we work with limited budget: active learning approaches must be executed only once and therefore it will be a waste to exhaustively implement all methodologies. Moreover, part of the budget might be used for setting up of a *hold-out set*. We note that in practice, it is advisable to first label the test base randomly in order to be as close as possible to the underlying distribution before labeling the training set.

- *When to stop the active learning process?* In addition to business aspects (e.g. limited budget or labeling time) there are many other possible stopping criteria: for example [ZH07] proposes the *Max-confidence* stopping criterion which comes to analyze the entropy measure of the pool-set. More specifically we stop the active learning process when the entropy of each unlabeled example becomes smaller than a given threshold (e.g. 0.001). The authors also propose the *Min-error* which consists in analyzing the performance of the learning model: we stop the active learning iterations when the models have reached a given performance.
- *Which learning models to use?* In active learning, most of the queried samples are adjusted to a given learning model. Since we do not know the labels in advance, the choice of the learning model remains delicate because at initialization the labeled data are not complex and abundant enough to choose the right learning model (a good fitted learning model at the beginning doesn't mean a good fit at the end of the active learning process).
- *What happens when labels are not reliable (mislabeling)?* The most common method is to re-label some (uncertain) data because of the possibility of errors in labelling the oracles.

2.5.2 Metrics and datasets

In order to evaluate the model performance we have to consider two components: an adequate model metric and a model unfairness evaluation.

Model precision. We use F1-score for evaluating the performance of a binary model on a hold-out set. F1-score measures the harmonic mean of *precision* (i.e. number of true positives divided by the total number of predicted positives) and *recall* (i.e. number of true positives divided by the total number of actual positives):

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}} \tag{2.6}$$

Model unfairness. The fairness of an estimator $h \in \mathscr{H}$ is measured on a test set $\mathscr{D}^{(test)}$ via the empirical counterpart of the unfairness measure $\mathscr{U}(h)$ given in Section 2.3.4. For simplicity's sake, we note $\hat{v}_{h|s}(k) = \frac{1}{|\mathscr{T}^s|}\sum_{(X,Y)\in\mathscr{T}^s} \mathbb{1}_{\{h(X)=k\}}$ the empirical distribution of $h(X)|S=s$ on $\mathscr{T}^s = \left\{(X,Y)\in\mathscr{D}^{(test)}|S=s\right\}$ the conditional test set. The unfairness is formally defined by:

$$\hat{\mathscr{U}}(h) = \frac{1}{K}\sum_{k=1}^{K}\left|\hat{v}_{h|-1}(k) - \hat{v}_{h|1}(k)\right|. \tag{2.7}$$

Let us now present the datasets.

Dataset. The textual data that we study is the NPS (Net Promoter Score) verbatims of Société Générale Insurance. The NPS is an indicator that measures the level of customer satisfaction. This indicator is measured through a survey carried out at regular intervals with a sample of customers.

Within the framework of the present study, the NPS analyzed concerns in particular the so-called "hot NPS"; this term expresses the fact that the survey is sent immediately to the customer following a call with a customer relations centers. The question asked to the customer to measure the level of satisfaction is the following:

"*Would you recommend our insurance company to a friend or your family?*"

The answer to this question is a score ranging from 1 (detractor) to 10 (promoter). These scores constitute our label space $\{1, \cdots, K\}$ with K a value between 2 (binary classification) and 10 (multi-class classification) depending on the split. We decided to study a binary classification framework $\mathscr{Y} = \{0,1\} = \{\text{score} \leq c, \text{score} > c\}$ with $c \in [K-1]$ being the *imbalanced* parameter. We can choose c such that we can range from a 10% (*imbalanced* case) to 50% (*balanced* case) imbalanced rate. In addition to the question the following question is also asked on surveys:

"*Why did you give this rating?*"

The answer to this question is a free text comment allowing clients to justify the rating given by explaining the main reasons for their satisfaction or dissatisfaction. This text comment is named "verbatim nps". It should be noted that a construction of a numerical representation model of verbatims in a "semantic space" allowing to measure the similarity of the verbatims has been implemented: for this purpose the algorithm *Doc2vec* is used [LM14]. Unlike embedding algorithms of type *Word2vec* [MSC+13], *Doc2vec* has the advantage of providing in a "native" way a digital representation of an entire textual document (here a verbatim for example) without having to resort to an aggregation of the embeddings of each of the words composing the verbatim. Each numerical representation of the textual documents represents our sample from $\mathscr{X}$. The sample made available for the active learning study consists of around 5000 verbatims that correspond to a collection period of approximately one year (late April 2018 to late May 2019).

Unfair dataset. The fairness is studied on two unfair datasets: one is UNPS studied in the main results, the other (LAW dataset) is in appendix A.

Unfair NPS (UNPS) dataset where we render the NPS dataset randomly unfair by introducing a sensitive feature S such that the distribution of S, given that $Y = k$, is

$$(S|Y=k) \sim 2 \cdot \mathscr{B}(p) - 1, \qquad \text{if} \quad k \leq \lfloor K/2 \rfloor$$

and

$$(S|Y=k) \sim 2 \cdot \mathscr{B}(1-p) - 1, \qquad \text{if} \quad k > \lfloor K/2 \rfloor$$

with p the parameter that measures the historical bias in the dataset and $\mathscr{B}$ the Bernoulli distribution. For instance, for $k \leq \lfloor K/2 \rfloor$, S takes the values 1 and -1 with probability p and $1-p$ respectively. Specifically, the model becomes fair when $p = 1/2$ and completely unfair when $p \in \{0,1\}$. We set $p = 0.7$.

2.5.3 Methods and settings

We will study the efficiency of active learning methods. The learning model used is XGBoost (*Extreme Gradient Boosting* proposed by [CG16]). This algorithm has the advantage of being flexible and is faster to train than Gradient Boosting. Thanks to a regularization term it adapts rather well to small or medium size data. We consider an uncertainty sampling with entropy measure (*EntropySampling*), a disagreement-based sampling with Query-by-bagging (*QbagSampling*), a model change sampling strategy with EGL (*EGLSampling*) and a density-based sampling (*DensityWeightedSampling*). The benchmark used here is a passive sampling (*RandomSampling*). In a nutshell, we will study active learning with respect to:

- the metric evaluation of the model to both balanced and imbalanced datasets;
- the unfairness evaluation of the *crude* model (without fairness filter) to both fair and unfair dataset;
- the unfairness evaluation of the *fair* model (with post-processed fairness filter) to both fair and unfair dataset. The fairness method is detailed in [DEHH21] and will be briefly presented below.

We split the data into three sets: 100 initial training set, 1000 test set and the remainder constitutes the pool set.

Fair multi-class classification. [DEHH21] provides an optimal fair classifiers with respect to the misclassification risk under Demographic Parity constraint. This method is called *argmax-fair*. The difficulty of obtaining an optimal fair classifier consists in finding a good equilibrium between misclassification risk and fairness criterion. Hence, given a hypothesis h, [DEHH21] introduces and calibrates the parameter $\lambda = (\lambda_1, \dots, \lambda_K) \in \mathbb{R}^K$ such that the so-called *fair-risk*

$$R_\lambda(h) := R(h) + \sum_{k=1}^{K} \lambda_k \left[\mathbb{P}(h(X) = k | S = 1) - \mathbb{P}(h(X) = k | S = -1)\right]$$

is minimized with respect to λ. The authors provide a plug in estimator for the optimal fair classifier h^*_{fair} with strong theoretical guarantees both in terms of fairness and risk. In particular, the fairness guarantee is distribution-free. Let us denote $(p^s_{t,x}(k))_k$ the conditional probabilities (*e.g.,* Random Forest, SVM, *etc.*) at time t for the instance x with the associated sensitive feature s calibrated by the training set $\mathscr{D}^{(train)}$. We also need the pool-set $\mathscr{D}^{(pool)}_{\mathscr{X}}$ to compute:

- $(\hat{\pi}_s)_{s \in \mathscr{S}}$ the empirical frequencies for estimating the distribution of the sensitive feature $\mathscr{S}$;
- N_s the number of observations corresponding to $S = s$. Therefore $N_{-1} + N_1 = \left|\mathscr{D}^{(pool)}_{\mathscr{X}}\right|$;
- and the feature vector in $\mathscr{D}^{(pool)}_{\mathscr{X}}$ denoted $X^s_1, \dots, X^s_{N_s}$ composed of *i.i.d.* data from $\mathbb{P}_{X^s}$, the distribution of $X^s := \{X | S = s\}$.

Given small perturbations $(\zeta_k)_{k\in[K]}$ and $(\zeta^s_{k,i})$ as independent copies of a uniform distribution on $[0,u]$ (e.g. $u = 10^{-5}$), the randomized fair classifier $\hat{h}^s$ by plug-in is

$$\hat{h}^s(x) = \underset{k\in[K]}{\operatorname{argmax}} \left(\hat{\pi}_s(p^s_{t,x}(k) + \zeta_k) - s\hat{\lambda}_k\right), \text{ for all } (x,s) \in \mathscr{X} \times \mathscr{S}$$

with $\hat{\lambda} \in \mathbb{R}^K$ given as

$$\hat{\lambda} \in \underset{\lambda}{\operatorname{argmin}} \sum_{s\in\mathscr{S}} \frac{1}{N_s} \sum_{i=1}^{N_s} \left[\max_{k\in[K]} \left(\hat{\pi}_s(p^s_{t,X^s_i}(k) + \zeta^s_{k,i}) - s\lambda_k\right)\right] .$$

[DEHH21] proposes to solve this optimization problem by smoothing the problem by soft-max (a.k.a. LogSumExp) and then use a gradient-based optimization method, such as accelerated gradient descent [Nes83, Nes13].

Note that as for all fairness-awareness algorithms, the downside of this method is that the model accuracy in predictions is poorer.

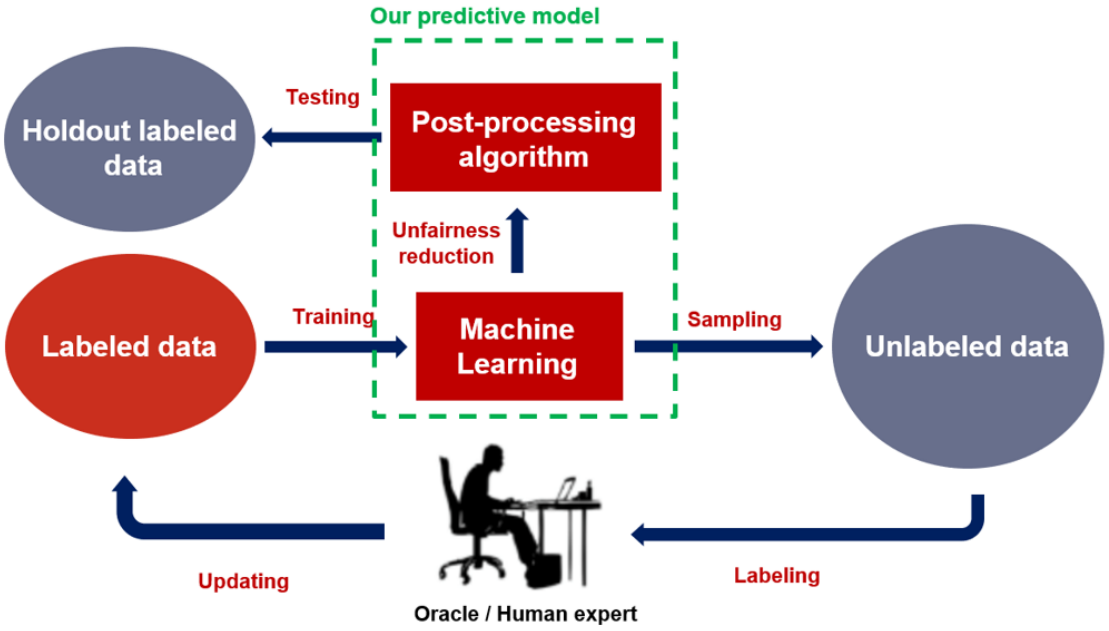


Figure 2.8: Fair active learning process: fairness-unawareness in queries.

2.5.4 Results

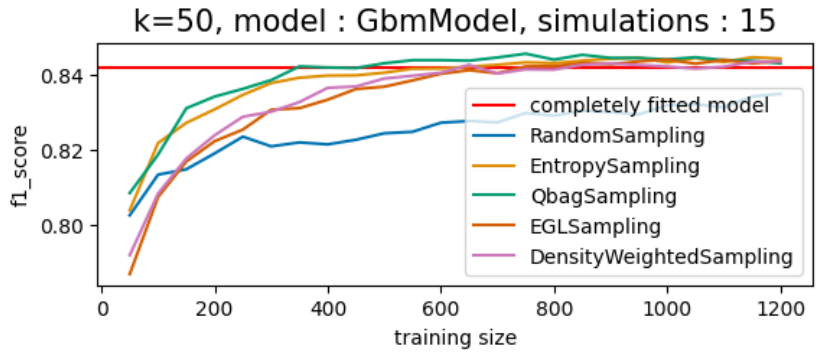
All the graphs presented in this section are averaged over 15 simulations and colored area corresponds to the standard deviation.

Model precision analysis. Numerical studies in Figures 2.9a and 2.9b show that active learning (AL) strategies outperform passive learning (PL) by sampling better quality data for the studied machine

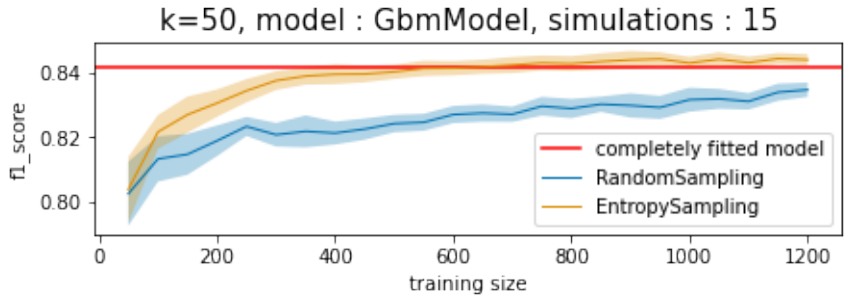
learning model (i.e. XGBoost). Indeed, most of the AL strategies converge to 0.84 with 600 labeled data instead of 2000 with PL. Figure 2.10 highlights it by evaluating the gap in performance between AL and PL:

$$\text{GAP} = 1 - \frac{\text{f1\_score\_passive}}{\text{f1\_score\_active}}$$

where f1_score_passive (resp. f1_score_active) is the F_1-score of the passive learning (resp. active learning) process. The gap in performance between AL and PL shown in Figure 2.10 indicates the efficiency of AL (the robustness to the imbalanced NPS can also be seen in Figure .12 in appendix A). Furthermore we note that, the more unbalanced the data, the closer the performance of AL is to the performance of PL (Figure .13 in appendix). Precisely [EHBG07] shows that AL performs well to a slightly imbalanced case but can be inefficient to a heavily imbalanced one.

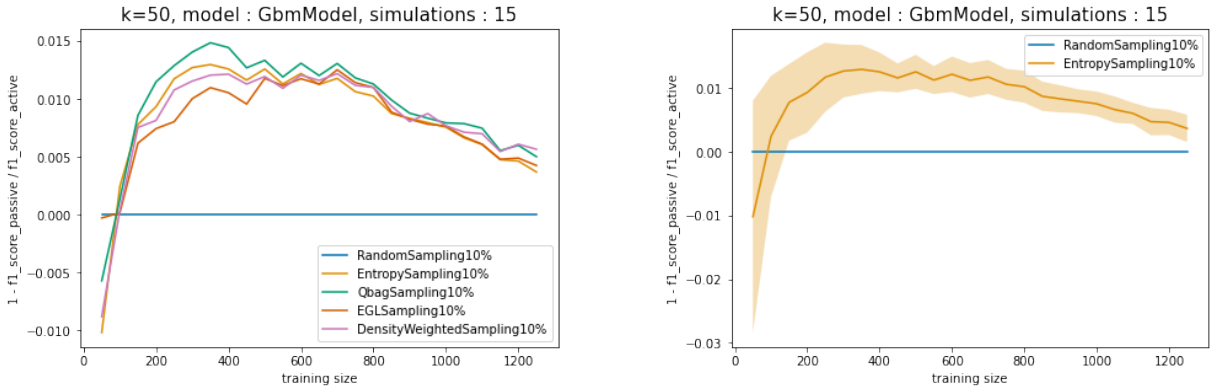


(a)



(b)

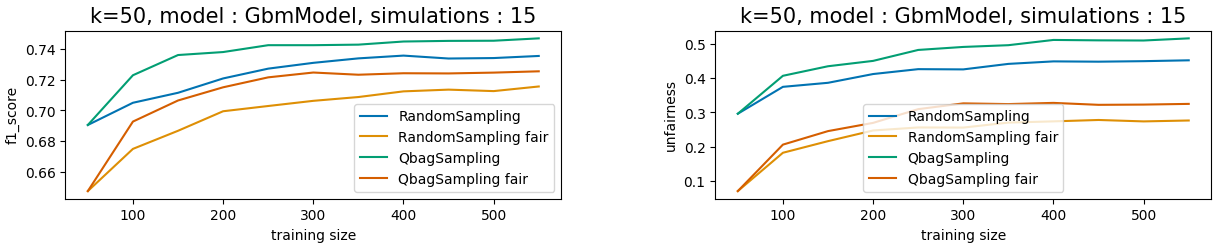
Figure 2.9: Performance of PL and AL on a balanced dataset
Performance of the classification procedures in terms of **F1-score** for the **XGBoost** estimator w.r.t the training set iteratively constructed by PL and AL methods mentioned above. In each iteration we query $k = 50$ instances on a **balanced** dataset (30%).



(a) (b)

Figure 2.10: Performance of PL and AL methods on a imbalanced dataset (10%). Performance of the classification procedures in terms of **F1-score** for the **XGBoost** estimator w.r.t the training set iteratively set by PL and AL methods on a **imbalanced** dataset (10%) where each iteration we query $k = 50$ instances. Each line corresponds to the mean over 15 simulations and the colored area the standard deviation (Figure (b)).

Model unfairness analysis. Figure 2.11 displays the metric (Figure 2.11a) and the fairness of both AL and PL w.r.t. training size. Figure 2.11b illustrates that PL is better than query-by-bagging method in terms of unfairness. However, the unfairness reduction algorithm proposed in this paper works well for both learning processes. We note that active learning processes work well for both crude and fair model. Thus active learning with post-processing fairness reduction algorithm seems a good trade-off between model accuracy and model unfairness. Interested readers are welcomed to see additional experimentations in appendix A.



(a) F1-score analysis (b) Unfairness analysis

Figure 2.11: Model performance on UNPS Dataset. Model performance (accuracy and unfairness) *w.r.t.* the training size. We study both fair and unfair model. Each line corresponds to the mean over 15 simulations and in each iteration we query $k = 50$ instances.

2.6 Conclusion

With the adoption of Artificial Intelligence, insurance companies are tied to the following objectives: (1) need of quality data alongside with an efficient learning model for deploying their AI solutions and (2) need of a good AI governance (interpretation and fairness) for managing and trusting their model. The present study introduces an overview of active learning for resolving these concerns: numerical analysis and application on real datasets show that active learning strategies can reduce considerably the amount of labelled data needed for calibrating an efficient machine learning model. In our study, the downside of unfiltered active learning seems that the model becomes more unfair. However it can be mitigated by adding a post-processing fairness such as *argmax-fair* leading to a good trade-off between model precision and unfairness. With its wide range of applications, fairness in multi-class classification is a rapidly expanding domain and we believe that considering both active learning and fairness will lead to enhance AI performance and mitigate operational and reputational risks in insurance companies.

Appendices

A Numerical experiments : additional figures

Model precision analysis.

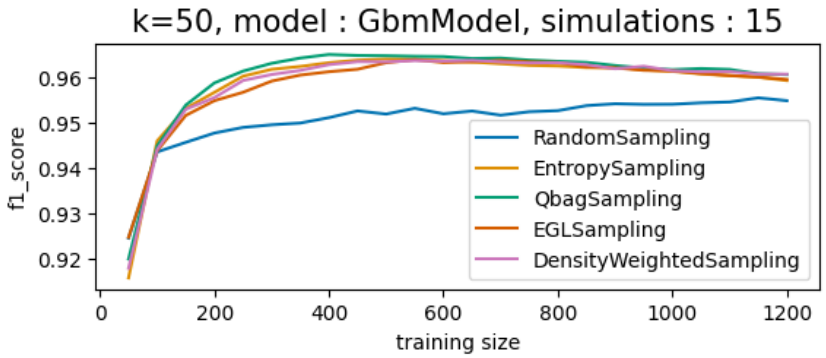


Figure .12: Performance of AL methods on a imbalanced dataset (10%) with PL as a baseline. Performance of the classification procedures in terms of **F1-score** for the **XGBoost** estimator w.r.t the training set iteratively constructed by *RandomSampling* and AL methods. We query $k = 50$ instances on a **imbalanced** dataset (10%).

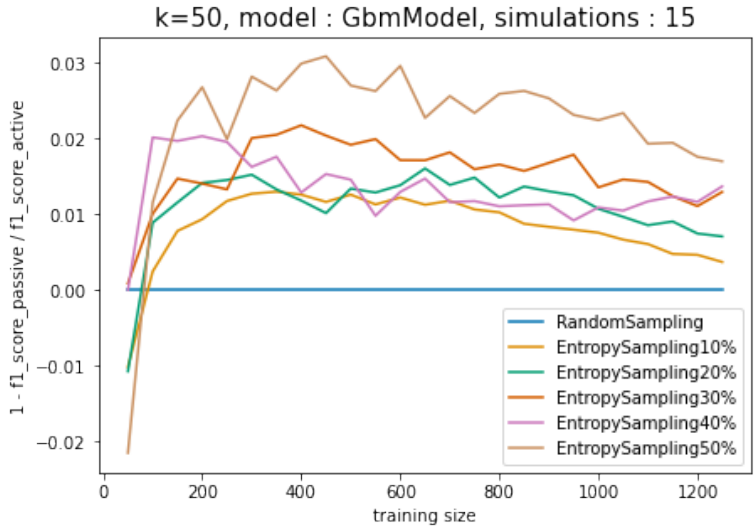
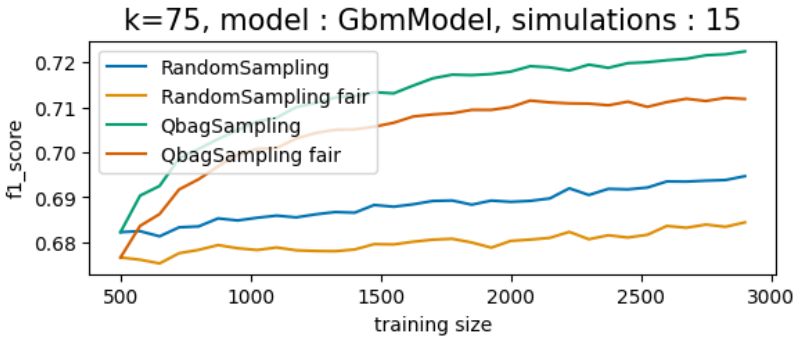


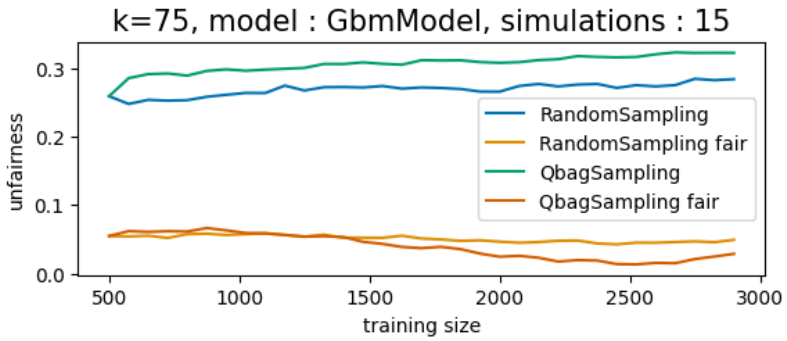
Figure .13: Performance of PL and AL methods on the various rates of imbalance (10% to 50%). Performance of the classification procedures in terms of **F1-score** for the **XGBoost** estimator w.r.t the training set iteratively set by passive learning and uncertainty-based methods. Each line corresponds to the mean over 15 simulations and in each iteration we query $k = 50$ instances w.r.t the dataset with different **rates of imbalance** (10% to 50%).

Model unfairness analysis. In the appendix, we examine a "real" unfair dataset and note that we obtain the same results as the UNPS dataset.

The studied dataset is *Law School Admissions* (LAW) dataset [WR98] which presents national longitudinal bar passage data and has 20649 examples. The task is to predict a students GPA divided into $K = 2$ classes such that the dataset is balanced. The sensitive attribute is the race (white versus non-white).



(a) F1-score analysis



(b) Unfairness analysis

Figure .14: Model performance on LAW Dataset.
Model performance (accuracy and unfairness) *w.r.t.* the training size. We study both unfair and fair model. Each line corresponds to the mean over 15 simulations and in each iteration we query $k = 75$ instances.

Chapter 3

Dynamic batch active learning

Contents

Active learning attempts to maximize a model's performance gain while annotating the fewest samples possible. This often suggests requesting as few labels as possible in each AL iteration. However, in practice, the delay in re-training the model makes it difficult to sample small batches of instances, as annotators must wait between each AL iteration. In particular, this problem becomes apparent when a complex learning model takes a long time to retrain. A trade-off must therefore be found between the performance of the model and the cost of the re-training time. The originality of this paper is considering a dynamic size of batch sequences (instead of the usual static size) and formulate an adequate stochastic optimization problem before optimizing it. To solve the optimal control problem, an approach based on the dynamic programming principle lead us to the Hamilton Jacobi Bellman equation solved by the value function.

Keywords: Batch mode active learning, stochastic control, dynamic programming.

3.1 Introduction

Over the last decades, with the democratization of the digitization, raw data are stored and becomes more and more abundant (e.g. emails, online customer reviews, videos, recordings,...). This democratization contributes greatly to the technological advances such as Artificial Intelligence (AI) solutions. These AI solutions are typically powered by fitting machine learning models to data. However the data usually comes unlabeled and recent learning algorithms are increasingly greedy in terms of labeled data (e.g. deep learning methods), therefore labeling them can bring considerable gains in predictive performances. In recent years, active learning (AL) process has become a growing interest [DE95, SC08b, HJZ14, GG15b, BST17, LBH18, WZS19] mainly due its ability to select the most "optimal" training set and avoid an important amount of labeling costs. Given the responses of a fitted model, one AL iteration consists in querying the labels of the best set of instances from one or several expert annotators or oracles. This results to a more accurate model as the training set is more informative. Often, during the AL iterations, the model quickly becomes more accurate than a passive approach (i.e. querying random instances) [TK01, Han09, FZTN+12]. As such, AL has been successfully employed in a number of applications, including text categorization [FZTN+12, GKBG18, AWH18, ZXX18], image categorization [HJZL06, WZL+16, GIG17, BGNK18] and speech recognition [NMS12, YCK19, AB19, LWCX21]. See [EHHJ21] for an overview of AL methods for insurance.

Traditionally each AL iteration is centered around selecting one instance to label before re-fitting the learning model to the whole labeled dataset. Labeling too much instances in a single AL iteration may reduce the quality of the human-machine interaction feedback loop. However in practice this framework raises issues of speed and adaptability. Indeed, a concern for AL is the amount of time human experts (or oracles) wait for the next instances to label: a more adaptable process would be to let annotators label several instances in a row. This issue becomes apparent when

- many oracles are available for a parallel annotation;

- the learning model is complex (*e.g.* deep learning models that require hours of retraining).

We call this problem the Cost of (retraining) Delay (or *CoD* for short). Note that CoD is inspired from *agile*[1] terminologies. In agile methodologies, CoD is a framework that helps a business to quantify the economic value of completing a project sooner as opposed to later [PMM18, Gol21]. As stated by [Rei09], "If you only quantify one thing, quantify the cost of delay.".

In AL, to overcome this CoD problem, a more suitable process is querying the label of a batch of instances at each AL iteration. This particular case of AL is called batch mode active learning (BMAL). We define *local* CoD the cost of retraining delay at the level of the AL iterations and *global* CoD the total cost of delay of the entire AL process. Equivalently for a given labeled budget, the global CoD problem amounts to reducing the number of AL iterations. In this context a well defined BMAL method enables to establish a trade-off between model performance and global CoD.

Related works. In the literature, there are three main BMAL approaches: (1) *the ranking methods* rank unlabeled instances, usually on the basis of informativeness or representativeness of the instances (they sometimes take into account the diversity of the instances), and select the best among them [SC00, Bri03, SR10]; (2) *the objective-driven methods* formulate and solve an optimization problem resulting to a batch of the most informative (or representative) instances [GS07, CBP14, CBS+15, YMN+15, WZS19]; (3) *the cluster-based methods* reduce the unlabeled space to a subset containing informative (usually uncertain according to the model responses) instances and select some representative ones [PB12, SS18].

Up to our knowledge, little work focuses on the sequence of optimal batch sizes. While some papers highlight the best size to set in advance and others optimize a batch size for a given AL iteration [CBP14, LGW18], none of the previous works consider the sequence of batch sizes as a parameter to be calibrated taking into account the trade-off between model performance and global CoD. Particularly, [CBP14] proposes an adaptive BMAL where, at each AL iteration, the batch size as well as the selection criteria are combined in a single optimization problem solved with gradient-based methods. Note that their method penalizes large AL batches, which is not optimal in practice (in terms of both local and global CoD problems). Our paper highlights the importance of the AL batch size both in quantifying the performance of the model and in reducing the global CoD (i.e. reducing the number of AL iterations). More precisely, our contribution is to formulate the optimization problem dynamically : we consider the BMAL as a stochastic control problem with the quality of the model as a stochastic process and the batch size as the control parameter. The total annotation budget (ie the number of data to be labeled) being fixed, our aim is to find the optimal sequence of batch sizes that maximizes a given criterion combining both the performance of the model and the cost of the labeling process.

Contributions. This paper focuses on the dynamic calibration of the optimal size of the queried instances. These sizes take account the quality of the model as well as the cost of retraining delay.

[1] Agile is a term used to describe approaches to software development including incremental delivery and team collaboration. Often in industry, agile is an iterative approach that helps teams deliver value to their customers faster.

More precisely,

1. we study the behaviour of BMAL methods on a real dataset ;
2. we sketch out the numerical analysis of BMAL with respect to the AL batch size ;
3. we formulate and resolve this problem as a Markov Decision Process framework using techniques of stochastic control and the dynamic programming principle (see e.g. [Kar81, Pha09]).

Outline. Section 3.2 studies the behaviour of BMAL methods w.r.t. the AL batch size on a real dataset. Section 3.3 formalizes the Batch Mode Active Learning methods as a decision process. Section 3.4 finds numerically the optimal sequence of AL batch sizes that maximizes a given criterion combining both the performance of the model and the cost of the labeling process.

3.2 Batch Mode Active Learning experiments

In this section, we study the behaviour of the BMAL procedure with different AL batch sizes.

3.2.1 Notations

A list of general notations classically presented in AL literature and introduced in the previous chapters is provided in Table 3.1. Given the mentioned notations, a *batch mode active learning* procedure consists in iteratively querying oracle to label a set of instances from $\mathscr{D}_{\mathscr{X}}^{(pool)}$ until an annotation budget B_{MAX} is exhausted. At each loop of the procedure (a.k.a. *AL iteration* or *time*), the training set $\mathscr{D}^{(train)}$ and the pool-set $\mathscr{D}_{\mathscr{X}}^{(pool)}$ are updated before retraining the learning model. We report the performance results of the learning model based on a hold-out set $\mathscr{D}^{(test)}$. A pseudo-code of the procedure is presented in Alg. 3. Note that the time t depends on the AL batch size (shorter time if the intermediate AL batch sizes are larger).

3.2.2 Dataset, model and metric

As explained in previous sections, the choice of the size b is crucial in the iterative performance of the model. Let us highlight this statement by analyzing it in depth on a toy dataset : the IMDb base [MDP+11], which is a collection of movie's reviews (our $X^i \in \mathscr{X}$) followed by the reviewer's sentiment (our $y \in \mathscr{Y}$ where $\mathscr{Y} = \{-1, 1\}$ with -1 for a negative sentiment and 1 for a positive sentiment). The original dataset is cleaned (such as removing HTML tags, stemming the words, ...) and tokenized before being embedded with *Doc2vec* [LM14]. This new tabular data is then fitted by a logistic regression. The metric used in this study is F1-score which measures the harmonic mean of *precision* (i.e. number of true positives divided by the total number of predicted positives) and *recall* (i.e. number of true positives divided by the total number of actual positives)

$$F_1 = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$

Notations	Descriptions
$\mathscr{X}$	the space of instances
$\mathscr{Y}$	the space of labels (or classes)
$\mathbb{P}$	the distribution over $\mathscr{X}$
$\mathscr{H}$	$\mathscr{H} : \mathscr{X} \to \mathscr{Y}$ the space of hypotheses
$\mathscr{D}^{(train)}$	$\mathscr{D}^{(train)} = \{(x_i^{(train)}, y_i^{(train)})\}_{i=1}^{L}$ the training set of size L
$\mathscr{D}^{(test)}$	$\mathscr{D}^{(test)} = \{(x_i^{(test)}, y_i^{(test)})\}_{i=1}^{T}$ the test set of size T
$\mathscr{D}_{\mathscr{X}}^{(pool)}$	$\mathscr{D}_{\mathscr{X}}^{(pool)} = \{x_1^{(pool)}, \ldots, x_U^{(pool)}\}$ the pool set of size U
B_t	the training size at t-th AL iteration
B_{MAX}	annotation budget

Table 3.1: General schema for notations

3.2.3 BMAL procedure

Usually the sampling strategy is either based on the outputs of the training model $\mathbb{P}(h_t(x_i) = y|x_i)$ capturing the model informativeness or based on the representativeness of instances w.r.t. the pool-set. For instance, in a batch-mode setting, the strategy proposed by [WZS19] combines the instances informativeness and representativeness in a single formulation. More specifically, at each AL iteration, [WZS19] proposes to sample the set of instances $\mathscr{D}_{\mathscr{X}}$ that verifies :

$$\min_{\mathscr{D}_{\mathscr{X}} \subset \mathscr{D}_{\mathscr{X}}^{(pool)}, |\mathscr{D}_{\mathscr{X}}| = b_t} R(\mathscr{D}_{\mathscr{X}}) + \lambda LC(\mathscr{D}_{\mathscr{X}})$$

where R is the representativeness score, LC is the lower-bounded certainty score and λ is the trade-off parameter between the scores R and LC. The particularity of LC is that we ignore the certainty score of some low certainty samples by introducing a lower bound on the certainty score of all instances in $\mathscr{D}_{\mathscr{X}}^{(pool)}$: for a set of instances $\mathscr{D}_{\mathscr{X}} \subset \mathscr{D}_{\mathscr{X}}^{(pool)}$, $\epsilon > 0$ and a certainty score C for one instance, a

Algorithm 3 Outline of the BMAL procedure

Input: h a base estimator, $\mathscr{D}^{(train)}$ the initial training set of size b_0 and $\mathscr{D}_{\mathscr{X}}^{(pool)}$ the initial pool-set.

At time $t = 0$, we set $B_0 = b_0$,

At time $t \geq 1$ (i.e. t-th AL iteration),

Step 1. Fit h on the training set $\mathscr{D}^{(train)}$

Step 2. Set the batch size b_t, an integer between 1 and $B_{\text{MAX}} - B_{t-1}$ and let the active learner query the label $y_1, \ldots, y_{b_t} \in \mathscr{Y}$ of the instances $x_1, \ldots, x_{b_t} \in \mathscr{D}_{\mathscr{X}}^{(pool)}$

Step 3. Update the training set and the pool-set:

$$\begin{aligned} \mathscr{D}^{(train)} &= \mathscr{D}^{(train)} \cup \{(x_1, y_1), \ldots, (x_{b_t}, y_{b_t})\} \\ \mathscr{D}_{\mathscr{X}}^{(pool)} &= \mathscr{D}_{\mathscr{X}}^{(pool)} - \{x_1, \ldots, x_{b_t}\} \\ B_t &= B_{t-1} + b_t \end{aligned}$$

Step 4. As long as we do not exhaust the labeling budget B_{MAX} (i.e. $B_t < B_{\text{MAX}}$), we repeat this process (**return to step 1**).

`Note that` B_t `is simply the size of` $\mathscr{D}^{(train)}$`.`

Output: the final estimator h

lower-bounded certainty score LC is defined by

$$LC(\mathscr{D}_{\mathscr{X}}) = \sum_{x \in \mathscr{D}_{\mathscr{X}}} \max(C(x), \epsilon)$$

For a probabilistic model we can consider the certainty score

$$C(x) = \max_y \mathbb{P}(h(x) = y | x)$$

and the representativeness score by the Mean Maximum Discrepency between

$$R(\mathscr{D}_{\mathscr{X}}) = \text{MMD}(\phi, X_{\mathscr{D}_{\mathscr{X}}^{(train)} \cup \mathscr{D}_{\mathscr{X}}^{(pool)}}, X_{\mathscr{D}_{\mathscr{X}}^{(pool)} - \mathscr{D}_{\mathscr{X}}})$$

where ϕ is a kernel mapping. For a more details of the MMD method, see [GBR+12].

3.2.4 Static-size BMAL

Fig. 3.1 describes different performances of a fixed batch size BMAL simulations, in terms of F1-score. Each curve corresponds to a simulation of a fixed batch size $b \in \{40, 60, 80, 100, 160, 200\}$. As expected, if b varies, the behaviour of the curves varies as well. First, we observe that for sufficiently small values of b (e.g. here $b < 100$) the performance of active learning is greater than the passive learning in every iterations. Second, we note that an active learning process with a too large batch size, i.e. $b \in \{160, 200\}$, generates a model less efficient that the other methodologies. Therefore, generating an optimal machine learning model in BMAL procedures implies calibrating the batch size b (not fixed in advance by the user).

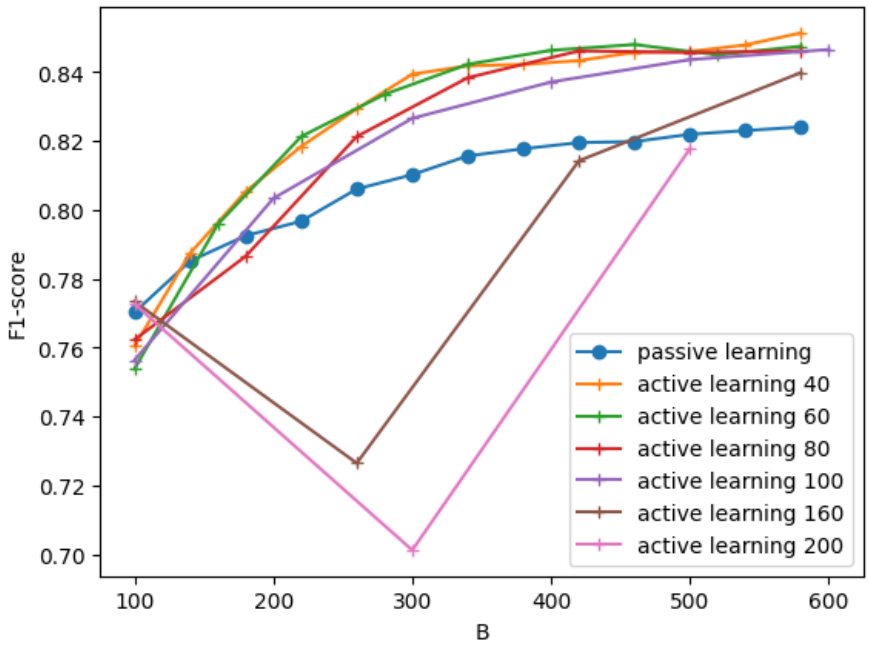


Figure 3.1: PL and static-size BMAL procedures with variable batch size. Each line correspond to the average process repeated over 15 simulations.

Furthermore, Fig. 3.2 shows that a smaller AL batch size implies better performance and lower volatility (see the difference between the boxplots in the top left figure and the boxplots in the bottom right figure).

3.2.5 Dynamic-size BMAL

In this section we try to intuit the dynamics of the batch size b_t where t is the t-th active learning iteration (i.e. time) by increasing (resp. decreasing) the batch size for each iteration.

Let denote by τ the first time where the consumed budget B hits B_{MAX}. If ever the performance of the model Q (e.g. F1-score) were able to reach 0 or 1, it would remain there,

$$\tau = \inf\{t \geq 0 \quad | \quad B_t = B_{MAX} \text{ or } Q_t = 0 \text{ or } Q_t = 1\}.$$

We denote the sequence of AL batch sizes by $\{b_0, \ldots, b_\tau\}$

Simulations. For each AL simulation, we have either increased or decreased the batch size by 20 per AL iteration. More precisely:

- by increasing the size of the batch at each time, we start with $b_0 = 140$ and end with $b_\tau = 20$
- whereas by decreasing the size of the batch at each time, we start with $b_0 = 20$ and end with $b_\tau = 160$.

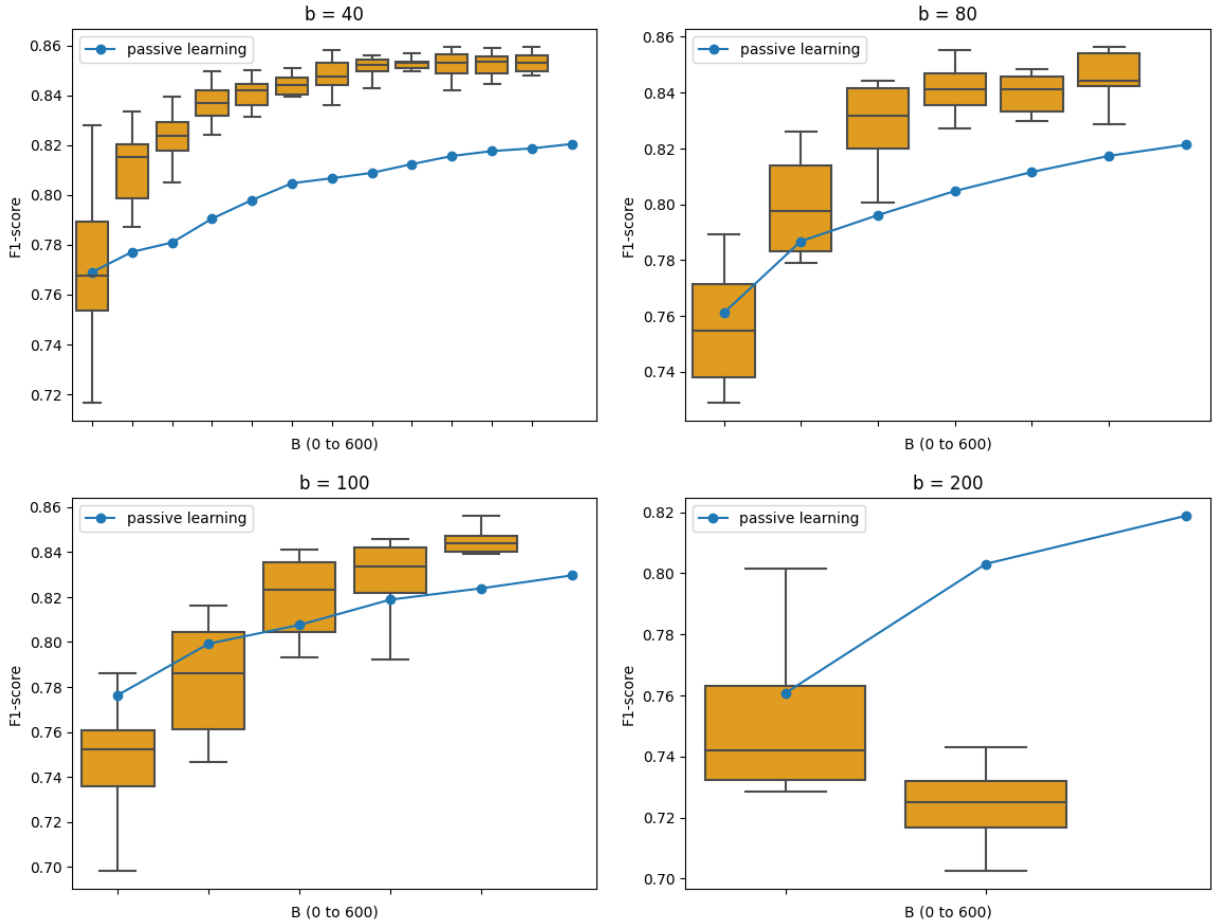


Figure 3.2: BMAL and PL procedures. The boxplots correspond to the BMAL process repeated over 15 simulations.

We call this type of strategy where we change the batch size during AL iterations the *dynamic-size* BMAL, as opposed to *static-size* BMAL the AL process with a fixed size strategy.

Empirical results. The results of this setting are shown in Fig. 3.3 where, as baseline, we consider the static-size BMAL with $b = 40$ and passive learning simulations. We note that:

- when b_t decrease with time ($b_0 = 140$ to $b_\tau = 20$), in the first iterations the performance can be worse than both passive learning and static-size BMAL, but as b_t becomes smaller and smaller, the performance reaches approximately the same performance as the static size sampling strategy. This result seems consistent with the following intuition: the learning model suffers from a cold start problem in the first iterations since the sampling is based on non-performing responses of this model (the training set does not contain enough informative instances yet).

- when b_t increase with time ($b_0 = 20$ to $b_\tau = 160$), the learning model seems to be as efficient as a static-size sampling strategy. In practice, this result is interesting because it shows that we can

significantly reduce the number of iterations in the active learning process needed to achieve the same result as static-size BMAL. Indeed, this can be a good start to solving the problem of having an active learning process with as few iterations as possible (thus less waiting time for the update of our learning model).

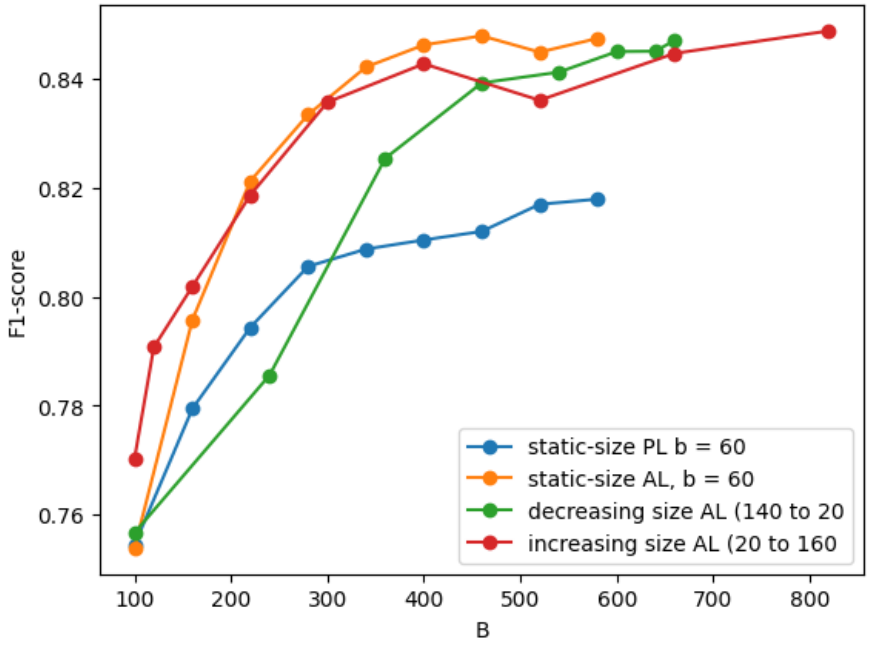


Figure 3.3: Comparison of BMAL procedures with static and "naïve" dynamic size over 15 simulations.

We keep in mind that, given a fixed budget of labelisation B_{MAX}, our objective is to find an optimal AL batch size sequence which maximizes the model performance while reducing the CoD. Thanks to these empirical results we have some insight into the behaviour of the volatility and the drift of the performance.

3.3 Batch Mode Active Learning with dynamic size

The AL procedure can be formalised as a decision process, i.e. a sequence of decisions where the process state corresponds to the AL iteration. This process, characterized by an action-state loop, can be viewed as a Markov Decision Process (MDP) where the outcomes are partly random and partly under the control of a decision maker. Each *action* of the process corresponds to the AL batch size in which we request its label and each *state* corresponds to the performance (or *quality*) of the model and the size of the training set. We end the decision process when we exhaust the annotation budget. The aim of this procedure is to maximise the total *reward* while reducing the intermediate *cost* once the budget is exhausted. The reward is the performance of the model and the cost is the CoD mentioned above. To solve this optimal control problem, an approach based on the dynamic programming principle

(DPP) is used. The DPP leads to a second-order non-linear partial differential equation (PDE), called the Hamilton Jacobi Bellman (HJB) equation associated with a value function.

3.3.1 Batch Mode Active Learning as a decision process

We recall that our objective is to find the optimal AL batch size sequence such that the procedure maximizes the model performance while reducing the intermediate CoD. In this section, we define this procedure as a MDP framework and show how the AL batch size policy can be calibrated from both labeled and unlabeled data. To benefit from the tractability of stochastic calculus, we formulate the problem in a continuous-time setting. We consider a filtered probability space $(\Omega, \mathbb{F}, \mathbb{P})$ where the filtration $\mathbb{F} = (\mathscr{F}_t)_{t\geq 0}$ is the natural filtration (complete and right-continuous) generated by a one-dimensional Brownian motion $(W_t)_{t\geq 0}$.

Markov Decision Process. A MDP framework is defined by $(\mathscr{S}, \mathscr{A}, \mathscr{T}, \gamma)$ with $\mathscr{S}$ the set of possible states, $\mathscr{A}$ the set of actions, $\mathscr{T}$ the transition function and γ the reward function.

State. At time t, the state processes (Q_t, B_t) are $\mathbb{F}$-adapted processes with values in $\mathscr{S}$. It corresponds to the couple of Q_t the performance (or quality) of the model and B_t the consumed budget. In multi-class classification the performance Q_t of the model is usually between 0 and 1. For example, we use the F_1-score for evaluating the performance of the model on the test set. The consumed budget $B_t = \int_0^{t\wedge\tau} b_s ds$ is a nonnegative real number between 0 (no queried instances) and B_{MAX} (exhaustion of the annotation budget), where b_t is the rate of labeling at time t and τ is the $\mathbb{F}$-stopping time

$$\tau = \inf\{t \geq 0 \quad | \quad B_t = B_{MAX} \text{ or } Q_t = 0 \text{ or } Q_t = 1\}.$$

.

Action or Control Process. At time t, the action $b_t \in \mathscr{A}$ corresponds to the rate of the AL batch's size, in which the active learning queries the label of the b_t instances in the pool-set. As an example of a sampling strategy (proposed by [WZS19]) in batch mode, we refer to section 3.2.3.

State processes. The performance process $(Q_t)_t$ is assumed to satisfy a Stochastic Differential Equation (SDE) driven by the Brownian motion $(W_t)_t$, whose coefficients (μ, σ) depend both on b_t the rate of annotated data and B_t the total number of annotated data. We set the dynamics of the state processes as

$$\begin{cases} dQ_t = \mu(B_t, b_t) \cdot Q_t(1-Q_t) \cdot dt + \sigma(B_t, b_t) \cdot Q_t(1-Q_t) \cdot dW_t \\ dB_t = b_t \cdot dt \end{cases} \tag{3.1}$$

Note that we need to set the μ and σ functions. Thanks to the empirical results of the section 3.2.5 we have some insight into the behaviour of the volatility σ and the drift μ. In particular, σ and μ should be

increasing with b. In Section 3.4.1 we propose a parameter settings to solve the optimization problem introduced below.

Reward and cost. We recall that our goal is to find the optimal AL batch size rate process $(b_t)_t$ which maximizes a (random) *reward* (e.g. the model performance) while reducing the intermediate (deterministic) labeling cost (e.g. CoD). As usual, the reward γ_t is a concave regular utility function U of Q_t:

$$\gamma_t = U(Q_t).$$

The utility function U models the risk-aversion of the user concerning the labeling-performance of the model. In the meantime, the cost is assumed to be a convex function $c : b \to \mathbb{R}$ of the batch size b. From these formalisations, we formulate our optimization problem as maximizing

$$\sup_{b \in \mathscr{A}} \mathbb{E}\left[U(Q_\tau) - \int_0^\tau c(b_s)ds\right]. \tag{3.2}$$

In Section 3.3.2 we solve the corresponding HJB equation leading us to an improved dynamic-size BMAL, and we provide some insightful numerical results in Section 3.4.

3.3.2 Dynamic Programming Principle and Hamilton Jacobi Bellman equation

In this section we compute the optimal AL batch size sequence resulting from the optimization problem Eq. (3.2).

3.3.2.1 The optimal feedback control

We recall the dynamics of the state processes (Q_t, B_t) in Eq. (3.1). We apply then the dynamic programming principle: DPP is an optimization method based on the following principle of optimality defined by Bellman [Bel58, Bel66] in the 1950s "An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision." Therefore DPP implies to consider the following value function (we refer to [Kar81, Pha09] for in depth-surveys of stochastic control and DPP)

$$v(Q_t, B_t) = \sup_{b_s,\, s \in [t,(t+h)\wedge\tau]} \mathbb{E}\left[v(Q_{t+h}, B_{t+h}) - \int_t^{t+h} c(b_s)ds \,\middle|\, \mathscr{F}_t\right]. \tag{3.3}$$

If ever Q were able to reach 0 or 1, it would remain there, so that the function v (if regular enough) should satisfy as well, for any $B \in [0, B_{MAX}]$

$$\begin{aligned} v(0^+, B) &= U(0) \\ v(1^-, B) &= U(1) \end{aligned}$$

We now apply Itô's formula for a given control b when $t+h \leq \tau$

$$\begin{aligned}
& v(Q_{t+h}, B_{t+h}) \\
= \; & v(Q_t, B_t) + \int_t^{t+h} \frac{\partial v}{\partial Q}(Q_s, B_s) dQ_s + \int_t^{t+h} \frac{\partial v}{\partial B}(Q_s, B_s) dB_s + \frac{1}{2} \int_t^{t+h} \frac{\partial^2 v}{\partial Q^2}(Q_s, B_s) d\langle Q \rangle_s \\
= \; & v(Q_t, B_t) + \int_t^{t+h} \left[\frac{\partial v}{\partial B}(Q_s, B_s) \sigma(B_s, b_s) Q_s (1 - Q_s) \right] dW_s \\
+ \; & \int_t^{t+h} \left[\mu(B_s, b_s) Q_s (1 - Q_s) \frac{\partial v}{\partial Q}(Q_s, B_s) + b_s \frac{\partial v}{\partial B}(Q_s, B_s) + \frac{1}{2} \sigma(B_s, b_s)^2 Q_s^2 (1 - Q_s)^2 \frac{\partial^2 v}{\partial Q^2}(Q_s, B_s) \right] ds
\end{aligned}$$

so we get

$$\begin{aligned}
& \mathbb{E}\left[v(Q_{t+h}, B_{t+h}) | \mathscr{F}_t \right] = v(Q_t, B_t) + \\
& \int_t^{t+h} \mathbb{E}\left[\mu(B_s, b_s) Q_s (1 - Q_s) \frac{\partial v}{\partial Q}(Q_s, B_s) + b_s \frac{\partial v}{\partial B}(Q_s, B_s) + \frac{1}{2} \sigma(B_s, b_s)^2 Q_s^2 (1 - Q_s)^2 \frac{\partial^2 v}{\partial Q^2}(Q_s, B_s) \middle| \mathscr{F}_t \right] ds
\end{aligned}$$

Hence, for any control b, we have

$$\begin{aligned}
& 0 \geq - \int_t^{t+h} \mathbb{E}\left[c(b_s) | \mathscr{F}_t \right] ds \\
+ \; & \int_t^{t+h} \mathbb{E}\left[\mu(B_s, b_s) Q_s (1 - Q_s) \frac{\partial v}{\partial Q}(Q_s, B_s) + b_s \frac{\partial v}{\partial B}(Q_s, B_s) + \frac{1}{2} \sigma(B_s, b_s)^2 Q_s^2 (1 - Q_s)^2 \frac{\partial^2 v}{\partial Q^2}(Q_s, B_s) \middle| \mathscr{F}_t \right] ds
\end{aligned}$$

Multiplying by $1/h$ and sending h to 0, we deduce:

$$0 \geq -c(b_t) + \mu(B_t, b_t) Q_t (1 - Q_t) \frac{\partial v}{\partial Q}(Q_t, B_t) + b_t \frac{\partial v}{\partial B}(Q_t, B_t) + \frac{1}{2} \sigma(B_t, b_t)^2 Q_t^2 (1 - Q_t)^2 \frac{\partial^2 v}{\partial Q^2}(Q_t, B_t)$$

This means that the drift of the value function is non-positive (hence the value function is a super-martingale) and it should be equal to zero (hence a martingale) for the optimal control b^*. Therefore, if we denote

$$A(B, b, Q) = \mu(B, b) Q (1 - Q) \frac{\partial v}{\partial Q}(Q, B) + b \frac{\partial v}{\partial B}(Q, B) + \frac{1}{2} \sigma(B, b)^2 Q^2 (1 - Q)^2 \frac{\partial^2 v}{\partial Q^2}(Q, B) - c(b)$$

the HJB equation in the interior of the domain $[0,1] \times [0, B_{MAX}]$ rewrites

$$\sup_{b \geq 0, b \leq B_{MAX} - B} \{ A(B, b, Q) \} = 0 \tag{3.4}$$

together with the boundary condition

$$v(Q, B_{\text{MAX}}) \quad = \quad U(Q) \quad \text{for} \quad Q \in (0,1) \tag{3.5}$$

The optimal control b^* as a function of (Q, B) is the optimizer in (3.4).

3.3.2.2 EDP approximation by finite difference and Howard algorithm

Let us numerically approximate the HJB equation.

Discretization. To discretise the equation Eq.(3.4), finite difference approximation methods can be used. The finite difference method consists in finding an approximation of the solution of a PDE at the "nodes" of a regular grid. More precisely, we discretize the interior of the domain $[0,1]\times[0,B_{\text{MAX}}]$ into a grid of $n_Q \times n_B$ equidistant points such that,

$$0 = Q_0 < Q_1 < \cdots < Q_{n_Q-1} = 1$$

and

$$0 = B_0 < B_1 < \cdots < B_{n_B-1} = B_{\text{MAX}}$$

Let $\Delta_Q = 1/n_Q$ and $\Delta_B = B_{MAX}/n_B$ be respectively the step of the interval $[0,1]$ and the step of the interval $[0,B_{\text{MAX}}]$. Then for all $(i,j) \in \{0,\ldots,n_Q-1\}\times\{0,\ldots,n_B-1\}$, $B_i = i\Delta_B$ and $Q_j = j\Delta_Q$. Let us denote $v_{i,j} = v(B_i,Q_j)$ and approximate A by:

$$\begin{aligned}\hat{A}(B,b,Q,v_{i-1,j}) &= \mu(B,b)\cdot Q(1-Q)\cdot\frac{v_{i,j+1}-v_{i,j}}{\Delta_Q}\\ &+ b\cdot\frac{v_{i,j}-v_{i-1,j}}{\Delta_B}\\ &+ \frac{1}{2}\cdot\sigma(B,b)^2\cdot Q^2(1-Q)^2\cdot\frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{\Delta_Q^2}\\ &- c(b)\end{aligned}$$

Therefore the discretisation of the HJB equation is given by

$$\sup_{b\geq 0, b\leq B_{MAX}-B}\{\hat{A}(B,b,Q,v_{i-1,j})\} = 0 \tag{3.6}$$

Here the problem is to find b^* realizing the above expression. It is easy to deduce the following system:

$$\begin{cases} b^* = \arg\sup_{b\geq 0, b\leq B_{MAX}-B}\{\hat{A}(B,b,Q,v_{i-1,j})\} \\ v_{i-1,j} = \frac{\Delta_B}{b^*}\left(\mu(B,b^*)Q(1-Q)\frac{v_{i,j+1}-v_{i,j}}{\Delta_Q} + \frac{1}{2}\sigma(B,b^*)^2Q^2(1-Q)^2\frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{\Delta_Q^2} - c(b^*)\right) \\ \qquad\qquad +v_{i,j} \end{cases}$$

Note that b^* should be dependant to $v_{i-1,\cdot}$.

Algorithm. For each (B,Q) we can use the *Howard algorithm* [How60] to deduce all the values of $v_{i,j}$ by *backward induction*. This algorithm consists in alternating two steps until a stopping criterion:

1. The equation obtained in Eq. (3.6) is solved by replacing the previously calculated (or initialized) optimal control b in the Bellman equation, giving us a (candidate) value function v
2. The (candidate) optimal control b is calculated, based on the (candidate) value function v, by maximizing $\hat{A}$.

For more details, we refer to the pseudo-code of this algorithm in Alg. 4.

Algorithm 4 Optimization using Howard algorithm

Initialisation. start with an initial value b^0. Compute the solution u^0 of $\hat{A}(B,b^0,Q,u^0)=0$ i.e.

$$u^0 = \frac{\Delta_B}{b^0}\left(\mu(B,b^0)Q(1-Q)\frac{v_{i,j+1}-v_{i,j}}{\Delta_Q}+\frac{1}{2}\sigma(B,b^0)^2Q^2(1-Q)^2\frac{v_{i,j+1}-2v_{i,j}+v_{i,j-1}}{\Delta_Q^2}-c(b^0)\right) + v_{i,j}$$

Step 1. given u^k, find b^{k+1} maximizing

$$\sup_{b\geq 0, b\leq B_{MAX}-B}\left\{\hat{A}(B,b,Q,u^k)\right\}$$

compute the solution u^{k+1} such that $\hat{A}(B,b^{k+1},Q,u^{k+1})=0$

Step 2. if $|u^{k+1}-u^k|<\epsilon$, then set $v_{i,j-1}=u^{k+1}$ else go to **Step 1**.

Parametrisation of the functions governing the models. The dynamics of the state processes (B,Q), and thus the optimal control, depends on the following functions:

- the *drift* function $\mu(B,b)$ to control the trend or growth rate in the quality of the labeling-model;
- the *volatility* function $\sigma(B,b)$, to control the variability of the labeling-performance ;
- the convex *cost* $c(b)$ which defines the cost CoD of labeling.
- the *utility* $U(Q)$ which defines the agent preferences according to the quality (or performance) of the model.

We propose hereafter some examples of such functions (in particular for μ and σ), in the light of our numerical experimentations. For U and c we take standard power functions. Note that the user may choose another adequate shape of functions than the one we propose in the following section (section 3.4).

3.4 Numerical analysis

In this section, we focus our discussion on some particular functions involved in the formulation of the optimization problem Eq.(3.4) before calibrating the parameters of the functions and finding the optimal sequence of batch sizes.

3.4.1 Calibration of the functions

Fig. 3.4 displays the functions used in our simulation.

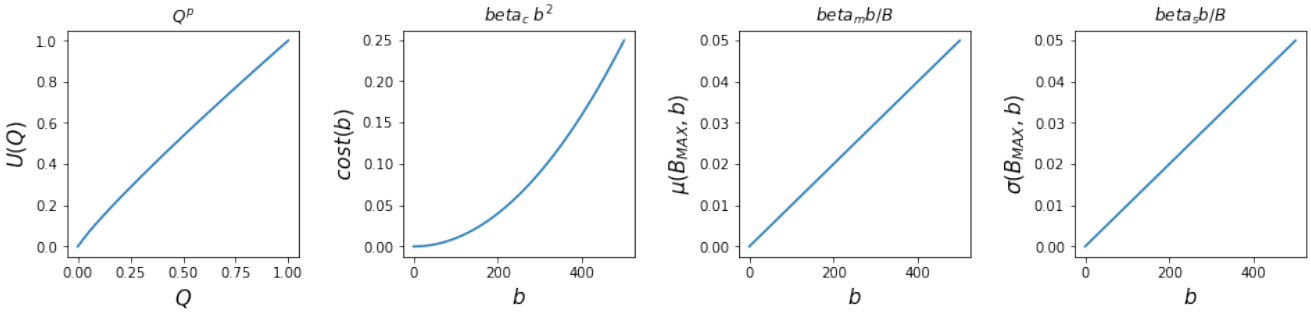


Figure 3.4: Functions defining the shape of the state process.

Utility function. The *utility* function U is taken as a standard utility function with Constant Relative Risk Aversion (CRRA)

$$U(Q) = Q^p$$

which is a concave function whose parameter $p \in (0,1)$ determines the relative risk aversion parameter (equal to $1-p$). Note that a smaller (resp. larger) value of p reflects a larger (resp. smaller) degree of risk aversion.

Cost function. The *cost* function C is defined as

$$C(b) = \beta_c \times b^2$$

with $\beta_c > 0$ a parameter calibrating the desired penalty for a too small batch size. This parameter can be different depending on the labeling situation (number of labelers, resources, ...).

Drift and volatility. We define the respectively *drift* function μ and the *volatility* function σ as follows

$$\mu(B,b) = \beta_m \times \frac{b}{B}$$

and

$$\sigma(B,b) = \beta_s \times \frac{b}{B}$$

with parameters $\beta_m, \beta_s > 0$.This means that both the drift and the volatility of the labeling-performance is increasing with respect to the rate of new labeled data over the total number of labeled data.

3.4.2 Adjusting parameters

Let us adjust the parameters $(\beta_c, \beta_m, \beta_s, p)$. For that purpose we have studied analytically the impact of these parameters on the behaviour of the BMAL procedure, reported in some heatmaps displaying the optimal AL batch size with respect to the state processes (Q, B).

Settings. We set $n_Q = 20$, $n_B = B_{\text{MAX}} = 500$ leading to $n_Q \times n_B = 1000$ grid points. We recall that thanks to the utility function the boundary conditions is well defined (see section 3.3.2):

$$\begin{aligned} v(0^+, B) &= U(0) \\ v(1^-, B) &= U(1) \\ v(Q, B_{\text{MAX}}) &= U(Q) \quad \text{for} \quad Q \in (0,1) \end{aligned}$$

Parameters. Fig. 3.5 to Fig. 3.8 show a sequence of heatmaps as a function of a parameter (four sequences of heat maps for four parameters). Each heatmap proposes an optimal AL batch size b^* given a couple (Q, B). In addition, interested readers may refer to Fig. .10 to .13 in the Appendix A for the rate of the optimal AL batch size $b^*/(B_{\text{MAX}} - B)$ in the remaining budget. Note that it might be more relevant to consider the rate $b^*/(B_{\text{MAX}} - B)$ rather than b^* because the problem depends on B_{MAX}. In these figures both at the beginning and at the end of labeling, our methodology tends to set small batch sizes (blue regions). Moreover, at the beginning of the labeling process, our methodology tends to label more when the quality of the model Q is either bad (Q close to 0 or $Q \approx 0$) or good (Q close to 1 or $Q \approx 1$). In BMAL procedures, this methodology is justified because:

- $(B \approx 0, Q \approx 0)$ the model is not qualitative enough to request efficiently the label of instances, so since the informativeness of each instance is difficult to interpret, reducing the CoD by proposing more labeling might be a good strategy.

- $(B \approx 0, Q \approx 1)$ the model is already well calibrated and "algorithmically" efficient so labeling more seems a good strategy.

Fig. 3.5 shows that for any couple (B, Q) a smaller value of β_c leads to labeling more during AL iterations. Fig. 3.6 and Fig. 3.7 shows that the methodology tends to label more when the drift β_m and/or the volatility β_s become large. In Fig. 3.8 the degree of concavity p corresponds to the degree of risk aversion of the methodology: The difference is most noticeable at the beginning of the labeling (i.e. $B \approx 0$). If $p \approx 1$, the amount of labeling of $Q \approx 0$ is approximately the same as that of $Q \approx 1$. As p becomes smaller, we become more risk averse, i.e. we label more for $Q \approx 1$ than for $Q \approx 0$.

Extensive experiments suggest that $(\beta_c, \beta_m, \beta_s, p) = (10^{-6}, 0.005, 0.05, 0.5)$ is an appropriate set of parameters.

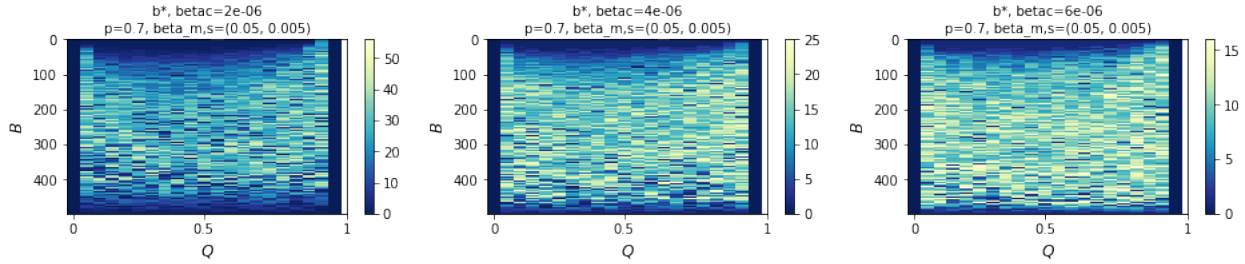


Figure 3.5: Sequence of Heatmaps (as a **function of** β_c) of the optimal control b^* w.r.t. the state process (B, Q)

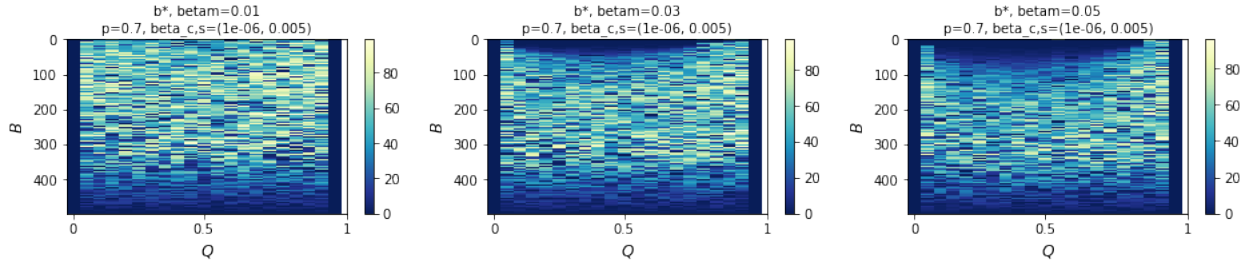


Figure 3.6: Sequence of Heatmaps (as a **function of** β_m) of the optimal control b^* w.r.t. the state process (B, Q)

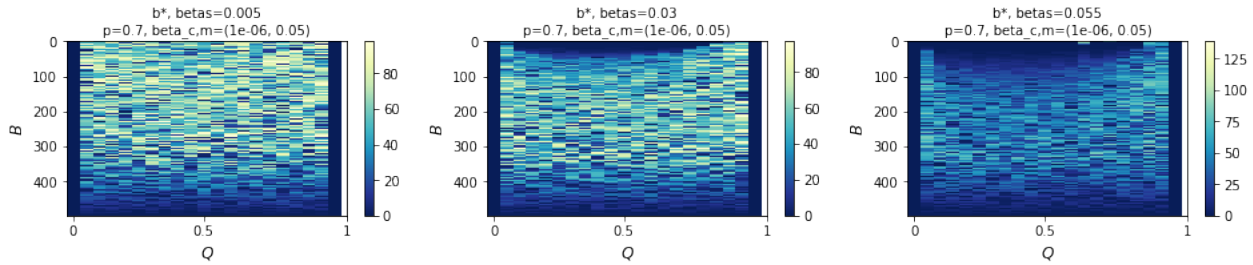


Figure 3.7: Sequence of Heatmaps (as a **function of** β_s) of the optimal control b^* w.r.t. the state process (B, Q).

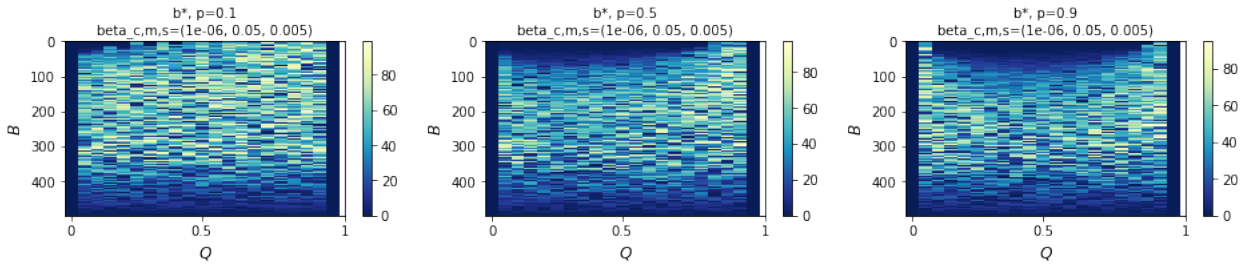


Figure 3.8: Sequence of Heatmaps (as a **function of** p) of the optimal control b^* w.r.t. the state process (B, Q).

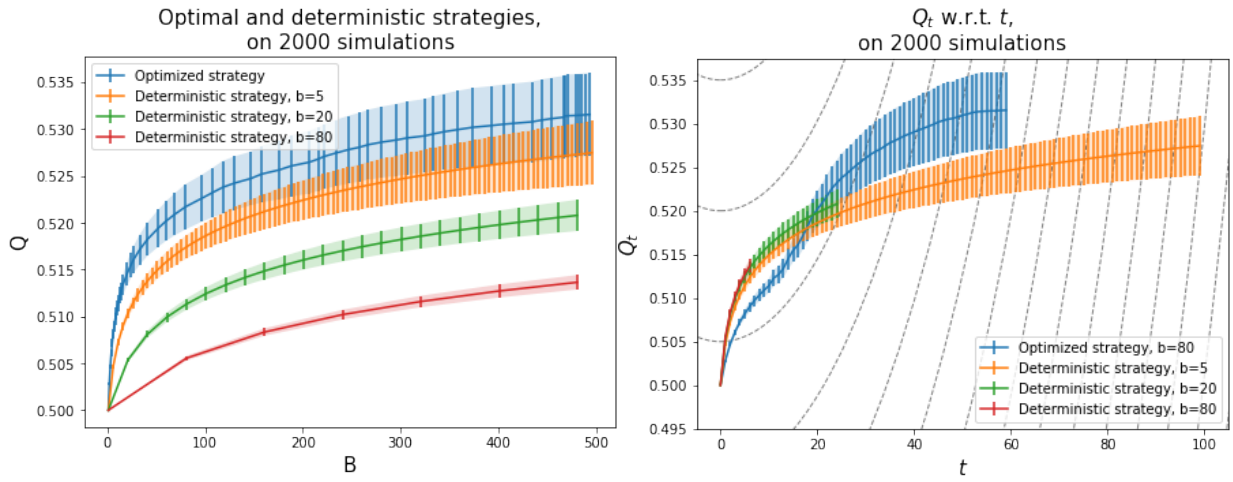
3.4.3 Numerical results

Let us simulate the optimized strategy (optimal policy calibrated by Alg.4) and compare it to some deterministic strategies (constant control). All simulations are based on the dynamics of the state processes defined in Eq. (3.1). Note that the dynamic (optimized) strategy and the deterministic strategy are "formulations" of the decision process of the dynamic-size BMAL and static-size BMAL procedures respectively.

Fig. 3.9 highlights the performance of the optimised strategies in terms of time and model quality. Fig. 3.9a shows that the optimized strategy outperforms the other deterministic strategies by labeling less at the beginning of the process to make the model more qualitative before moving to larger batch sizes (consistent with the BMAL experiments in section 3.2). Interested readers may refer to the heatmaps of Fig. .15 in Appendix A for an overview of the overall strategy. This result can also be noted on the value functions in Tab. 3.2. This dynamic strategy enables to reduce considerably the number of iterations. Indeed as shown by the Fig. 3.9b, given the budget B_{MAX} the dynamic strategy takes less time than the static strategy while being more qualitative.

Strategy	$Q_0 = 0.3$	$Q_0 = 0.5$	$Q_0 = 0.7$	$Q_0 = 0.9$
Optimized	0.577 ± 0.01	0.756 ± 0.007	0.867 ± 0.001	0.959 ± 0.0
Deterministic with $b = 5$	0.574 ± 0.001	0.753 ± 0.011	0.867 ± 0.001	0.959 ± 0.0
Deterministic with $b = 20$	0.574 ± 0.0	0.728 ± 0.006	0.867 ± 0.003	0.959 ± 0.0
Deterministic with $b = 80$	0.574 ± 0.0	0.727 ± 0.0	0.842 ± 0.0	0.95 ± 0.01

Table 3.2: Value functions w.r.t the control strategy. We report the means and standard deviations over the 2000 repetitions. Colored values highlight the best strategy.



(a) (b)

Figure 3.9: Optimal and deterministic strategies
Comparison between the optimised strategy and the deterministic strategies with the static batch size $b \in \{5, 20, 80\}$. Initially we set $(B_0, Q_0) = (0, 0.5)$. All simulations are over 2000 simulations and the colored areas correspond to the standard deviations. (a) Displays the average trajectories of the (B, Q) state process of the optimized and deterministic strategies and (b) displays the average quality w.r.t. the time (or iteration) of the optimized and deterministic strategies.

3.5 Conclusion

The study on the dynamic-size BMAL highlights the importance of the choice of the sequence of batch sizes. We have formulated this optimisation problem in a dynamic way. Numerical studies present an optimal labeling strategy that significantly reduces the number of AL iterations while maintaining good model performance. This would improve the labelling conditions for human experts.

However, this methodology faces certain difficulties. First is the calibration of the functions parameters. Indeed, in the present study we have provided these parameters using some heuristics based on numerical experiments. This question must of course be studied and refined. Second difficulty this methodology is complicated to use in practice on new databases. One methodology would be to learn/calibrate the optimal batch size policy from the data. This policy can then be generalised to other databases, making it "reusable". The aim is to be able to transfer this policy to other BMAL procedures

Appendices

A Additional experiments

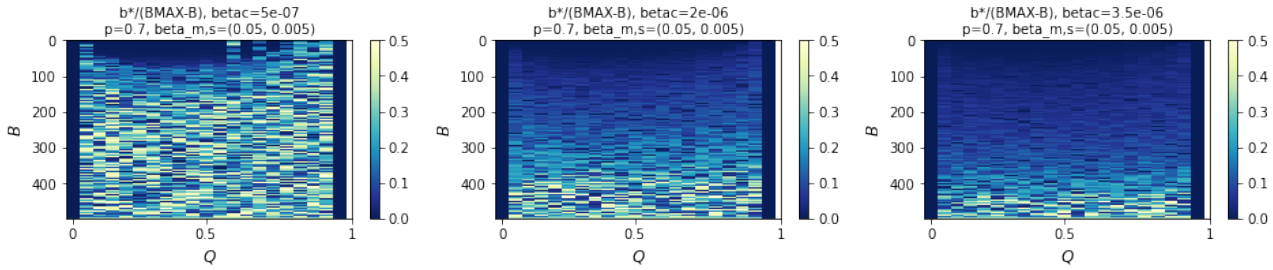


Figure .10: Rate of optimal control in the remaining budget $b^*/(B_{\text{MAX}} - B)$ w.r.t. (B, Q) (function of β_c)

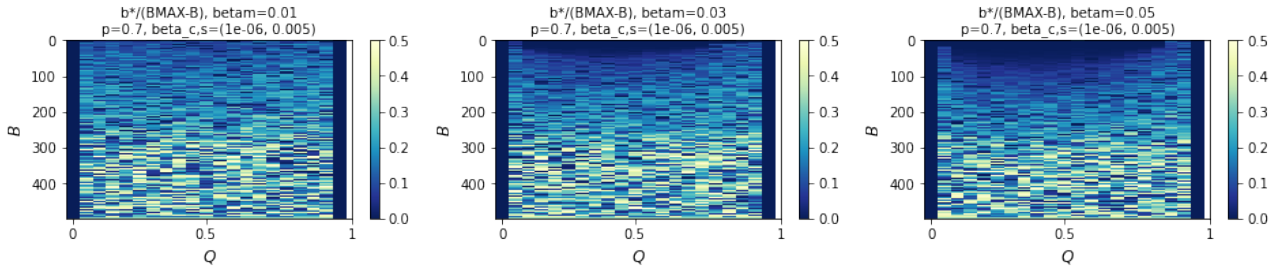


Figure .11: Rate of optimal control in the remaining budget $b^*/(B_{\text{MAX}} - B)$ w.r.t. (B, Q) (function of β_m)

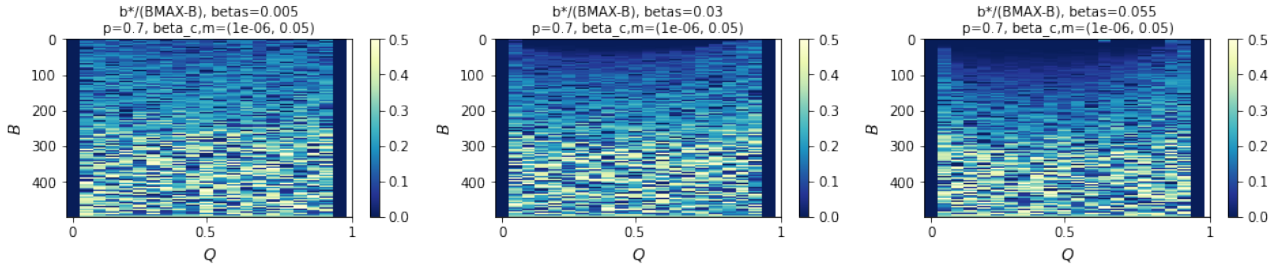


Figure .12: Rate of optimal control in the remaining budget $b^*/(B_{\text{MAX}} - B)$ w.r.t. (B, Q) (function of β_s)

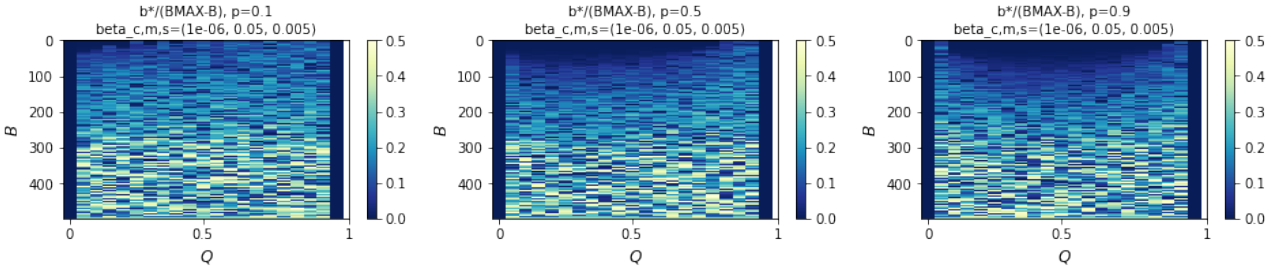


Figure .13: Rate of optimal control in the remaining budget $b^*/(B_{\text{MAX}} - B)$ w.r.t. (B, Q) (function of p)

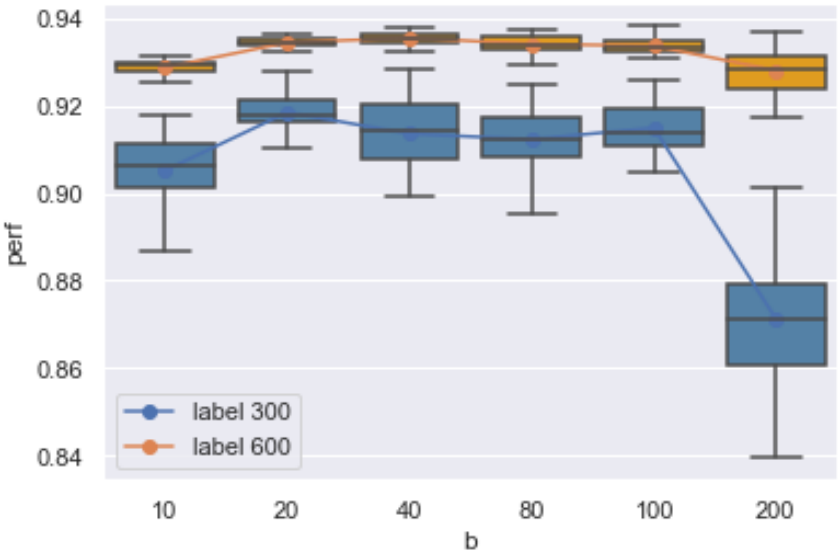
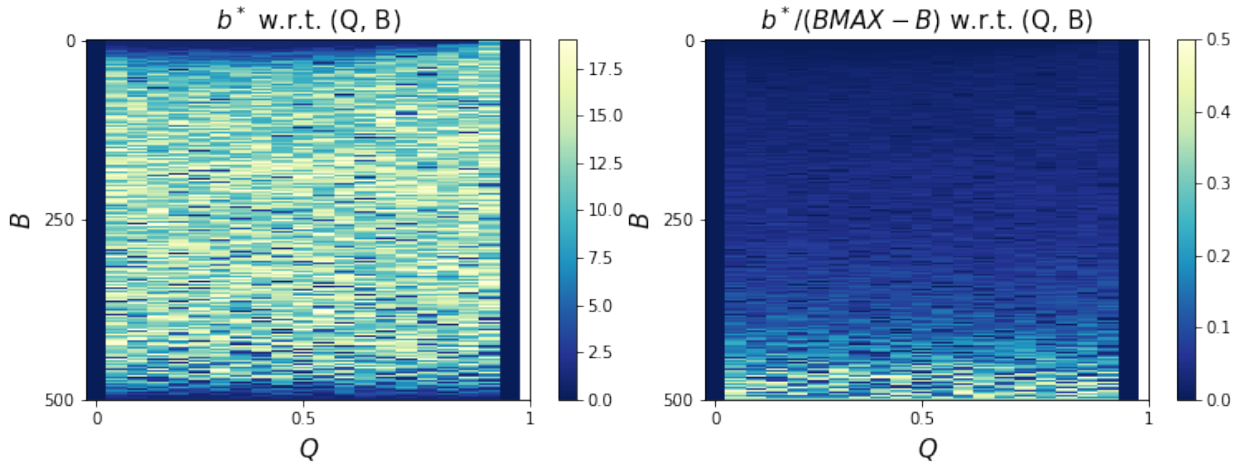


Figure .14: Performance of the model for a given fixed batch size in static-size BMAL. Each curve corresponds to a "cut" at a given label B. More precisely, for $b = 10$, the point of the curve "label 300" corresponds to the performance at $B = 300$ of a model built in a BMAL of batch size 10.



(a) (b)

Figure .15: Heatmaps of "X" w.r.t. the state process (B, Q) (a) X: the optimal control b^* (b) X: the rate of the optimal control in the remaining budget $b^*/(B_{\text{MAX}} - B)$

Chapter 4

Fairness guarantee in multi-class classification

Contents

Algorithmic Fairness is an established area of machine learning, willing to reduce the influence of biases in the data. Yet, despite its wide range of applications, very few works consider the multi-class classification setting from the fairness perspective. We extend both definitions of exact and approximate fairness in the case of *Demographic Parity* to multi-class classification. We specify the corresponding expressions of the optimal fair classifiers. This suggests a plug-in data-driven procedure, for which we establish theoretical guarantees. The enhanced estimator is proved to mimic the behavior of the optimal rule both in terms of fairness and risk. Notably, fairness guarantees are distribution-free. The approach is evaluated on both synthetic and real datasets and turns out to be very effective in decision making with a preset level of unfairness. In addition, our method is competitive with the state-of-the-art in-processing fairlearn in the specific binary classification setting.

Keywords: Algorithmic fairness, multi-class classification, statistical learning.

4.1 Introduction

Algorithmic fairness has become very popular during the last decade [ZWS+13, LJ16, CKP09, ZVGRG17, ADW19, ABD+18a, DOBD+18, CDH+19, CJS+20, BHN18] because it helps addressing an important social problem: mitigating historical bias contained in the data. This is a crucial issue in many applications such as loan assessment, health care, or even criminal sentencing. The common objective in algorithmic fairness is to reduce the influence of a sensitive attribute on a prediction. Several notions of fairness have already been considered in the literature for the binary classification problem [ZVGRG19, BHN18]. All of them impose some independence condition between the sensitive feature and the prediction. In some applications (e.g. loan agreement), this independence is desired on some or all values of the label space, see *Equality of odds* or *Equal opportunity* [HPS16]. In this paper, we focus on the well established *Demographic Parity* (DP) [CKP09] that requires the independence between the sensitive feature and the prediction function, while not relying on labels. DP has a recognized interest in various applications, such as for loan agreement without gender attributes or for crime prediction without ethnicity discrimination [HDFMB11, KZC13, BS14, FFM+15].

All the previously mentioned references focus either on the regression or the binary classification frameworks. However, many (modern) applications fall within the scope of multi-class classification, *e.g.,* image recognition, text categorization. Moreover, most real world applications can be tackled from the multi-class perspective. For instance, criminal recidivism is often treated as a binary problem, while it may be more suitable to distinguish between the different stratum of the problem and provide thinner description of the criminal behavior. However, extension of previous works to the multi-class setting is tricky, in particular since the adequate notion of fairness in this framework is not clearly defined and should be handled with caution.

Up to our knowledge, imposing fairness constraint in the multi-class problem has only been briefly discussed in [YX20], while focusing on Support Vector Machine (SVM) fair prediction. However, their fairness approach relies on properly selecting a subset of the data that is unbiased from the fairness perspective, and hereby differs significantly from the one presented here. We aim at enforcing fairness using the dataset as a whole! Besides, from a high-level perspective, the procedure described in [YX20] chooses to impose fairness on each component of the score function. It is clear that such methodology can be generalized to any convex empirical risk minimization (ERM) problem such as SVM or quadratic risk. However, since the decision rule in the multi-class setting relies on the maximizer over scores, we do not adopt this quite unnatural approach and rather directly impose fairness on the maximizer itself. Our main contributions are the following:

- We extend DP notion of exact and approximate fairness to the multi-class classification problem;
- We give optimal solutions for the multi-class classifier problem under exact or approximate DP constraints;
- We build a data-driven procedure that mimics the performance of the optimal rule both in terms of risk and fairness. Notably, our fairness guarantees are *distribution-free*;
- The robustness of our approach is illustrated on synthetic datasets with various bias levels, as well as on several real datasets. It proves to be very effective for decision making with a preset level of unfairness.

Related works. There are mainly three ways to build fair prediction: i) *pre-processing* methods mitigate bias in the data before applying classical Machine Learning algorithms; ii) *in-processing* methods reduce bias during training; iii) *pro-processing* methods enforce fairness after fitting. The present work falls within the last category. In a related study, [CDH+19] exhibit fair binary classifiers under *Equal Opportunity* constraints. In contrast, we focus on the multi-class setting, while imposing *DP* constraints.

Another line of works considers algorithmic fairness from an optimal transport perspective [CJS+20, GLR20, CDH+20c, GDBFL19]. A fair prediction is built upon Wasserstein barycenters of conditional distributions with respect to the sensitive feature. Such argumentation extends with little effort to a multi-class setting, even though the proper fairness definition in this context remains a question to investigate. This approach provides a fair classifier based on empirical risk minimization (ERM) with fairness constraints on each underlying score. However, fairness constraints on scores do not properly translate to fairness at the level of the classifier in the multi-class setting. Hence, we opt in the present work to enforce fairness directly at the level of the classifier.

Up to our knowledge, only few works study fairness in the multi-class setting. As previously detailed, [YX20] enforces fairness by sub-sample selection and is in-processing. In contrast, we keep the whole sample and enforce fairness in a post-processing manner. The multi-class framework is also considered in [TRN20]. However, the authors do not provide an explicit formulation of the optimal

fair rule. Furthermore, their theoretical fairness guarantee is not distribution free. Finally, they only consider numerical experiments for binary classification. Our method definitely provides valuable benefits on all these aspects.

Outline of the chapter. In the context of *exact* fairness, Section 4.2 defines DP fairness in the multi-class setting and explicit expression of the optimal fair classifier are provided. The corresponding data-driven procedure together with its statistical guarantees on risk and fairness are presented in Section 4.3. Section 4.4 extends the previous study to the case of *approximate fairness*. Section 4.5 details the algorithm implementation in the general approximate fairness setting, while numerical experiments are provided in Section 4.6.

4.2 Exact fairness in multi-class classification

This section focuses on exact fairness in multi-class classification. In particular, the optimal classifier for multi-class classification with *exact* DP constraint is established.

Let (X,S,Y) be a random tuple with distribution $\mathbb{P}$, where $X \in \mathcal{X}$ a subset of $\mathbb{R}^d$, $S \in \mathcal{S} := \{-1,1\}$, and $Y \in [K] := \{1,\dots,K\}$ with K being a fixed number of classes. The distribution of the sensitive feature S is denoted by $(\pi_s)_{s\in\mathcal{S}}$, and we assume that $\min_{s\in\mathcal{S}} \pi_s > 0$. A classification rule g is a function mapping $\mathcal{X} \times \{-1,1\}$ onto $[K]$, whose performance is evaluated through the misclassification risk

$$\mathcal{R}(g) := \mathbb{P}(g(X,S) \neq Y) \quad .$$

For $k \in [K]$, we denote $p_k(X,S) := \mathbb{P}(Y = k|X,S)$. Recall that a Bayes classifier minimizing the misclassification risk $\mathcal{R}(\cdot)$ over the set $\mathcal{G}$ of all classifiers is given by

$$g^*(x,s) \in \arg\max_k p_k(x,s), \quad \text{for all } (x,s) \in \mathcal{X} \times \mathcal{S} \quad .$$

4.2.1 Multi-class classification with demographic parity

We consider here DP constraint [CKP09], that requires independence of the prediction function from the sensitive feature S. Let first extend this notion to multi-class classification in the case of hard (exact fairness) constraints

Definition 4.1: Exact Demographic Parity

A classifier $g \in \mathcal{G}$ is *exactly* fair (denoted $g \in \mathcal{G}_{fair}$) with respect to the distribution $\mathbb{P}$ on $\mathcal{X} \times \mathcal{S} \times [K]$ if, for each $k \in [K]$,

$$\mathbb{P}(g(X,S) = k|S = 1) = \mathbb{P}(g(X,S) = k|S = -1) \quad .$$

This definition naturally extends the DP constraint considered in binary classification [ADW19, CJS+20, GDBFL19, JPS+19, ODP19]. When fairness comes into play, two important aspects of a classifier need to be assessed: the misclassification risk $\mathcal{R}(\cdot)$ and the unfairness, quantified as follows.

Definition 4.2: Unfairness measure

The unfairness of a classifier $g \in \mathscr{G}$ is quantified by

$$\mathscr{U}(g) := \max_{k\in[K]} |\mathbb{P}(g(X,S)=k|S=1) - \mathbb{P}(g(X,S)=k|S=-1)| \ .$$

Naturally, taking into account the definition above, a classifier g is *exactly fair* if and only if $\mathscr{U}(g)=0$.

Alternative measures of unfairness could be considered. The maximum can for instance be replaced by a summation over k. While both measures have their advantages, picking the maximum simplifies fairness evaluation in empirical studies.

4.2.2 Optimal exactly fair classifier

This section provides an explicit formulation of the optimal fair classifiers *w.r.t.* the misclassification risk under DP constraint. An optimal exactly fair classifier g^*_{fair} solves

$$\min_{g\in\mathscr{G}_{fair}} \mathscr{R}(g) \ .$$

Obtaining an optimal fair classifier requires to properly balance the misclassification risk together with the fairness criterion. For this purpose, let consider the Lagrangian of the above problem and introduce for $\lambda = (\lambda_1,\dots,\lambda_K) \in \mathscr{R}^K$,

$$\mathscr{R}_\lambda(g) := \mathscr{R}(g) + \sum_{k=1}^{K} \lambda_k [\mathbb{P}(g(X,S)=k|S=1) - \mathbb{P}(g(X,S)=k|S=-1)] \ . \tag{4.1}$$

We call this measure *fair-risk* and detail below how minimizing this risk for a properly chosen λ gives the fair classifier g^*_{fair}. We require besides the following technical condition.

Assumption 4.1: Continuity assumption

The mapping $t \mapsto \mathbb{P}\left(p_k(X,S) - p_j(X,S) \le t | S=s\right)$ is assumed continuous, for any $k,j \in [K]$ and $s \in \mathscr{S}$.

Assumption 4.1 implies that the distribution of the differences $p_k(X,S) - p_j(X,S)$ has no atoms. It is required to derive a closed expression of g^*_{fair}. Although it may sound unusual, it simplifies to the continuity of $t \mapsto \mathbb{P}(p_k(X,S) \le t|S=s)$ considered in [CDH+19] for the binary case ($K=2$). These conditions however describe different sets of distributions when $K \ge 3$. Assumption 4.1 is a tailored condition for the multi-class problem.

We are now in a position to provide a characterization of optimal fair classification.

Proposition 4.1

Let Assumption 4.1 be satisfied and define

$$\lambda^* \in \arg\min_{\lambda \in \mathbb{R}^K} \sum_{s \in \mathcal{S}} \mathbb{E}_{X|S=s} \left[\max_k (\pi_s p_k(X,s) - s\lambda_k) \right] .$$

Then, $g^*_{fair} \in \arg\min_{g \in \mathcal{G}_{fair}} \mathscr{R}(g)$ if and only if $g^*_{fair} \in \arg\min_{g \in \mathcal{G}} \mathscr{R}_{\lambda^*}(g)$.

In other words, the optimum of the risk $\mathscr{R}(g)$ over the class of fair classifiers also maximizes the fair-risk $\mathscr{R}_{\lambda^*}$. By construction, $\mathscr{R}_{\lambda^*}$ is a risk measure which efficiently balances both classification accuracy and unfairness. Proposition 4.1 directly implies that $\mathscr{R}_{\lambda^*}(g) \geq \mathscr{R}_{\lambda^*}(g^*_{fair}) = \mathscr{R}(g^*_{fair}) \geq 0$, for all $g \in \mathcal{G}$. Furthermore, Prop. 4.1 entails a closed form expression of optimal exactly fair classifiers, which is the bedrock of our procedure: any optimal fair classifier is simply maximizing scores, which are obtained by shifting the original conditional probabilities in a optimal manner.

Corollary 1

Under Assumption 4.1, an optimal *exactly fair* classifier is characterized by

$$g^*_{fair}(x,s) \in \arg\max_k \left(\pi_s p_k(x,s) - s\lambda^*_k \right), \ (x,s) \in \mathcal{X} \times \mathcal{S}.$$

Remark 4.1

The binary setting corresponds to the case where $K = 2$ (with label space $\mathcal{Y} = \{0,1\}$). In this specific setting, the fairness constraint reduces to a single constraint and the optimal rule from Corollary 1 simplifies as

$$g^*_{fair}(x,s) = \mathbb{1}_{\{p_1(x,s) \geq \frac{1}{2} + \frac{s\lambda^*}{2\pi_s}\}}, \quad (x,s) \in \mathcal{X} \times \mathcal{S} \ ,$$

where λ^* is solution in λ of

$$F_1\left(\frac{\lambda + \pi_1}{2\pi_1}\right) = F_{-1}\left(\frac{-\lambda + \pi_{-1}}{2\pi_{-1}}\right) \ ,$$

with $F_s(t) = \mathbb{P}(p_1(X,S) \leq t \mid S = s)$.

4.3 Data-driven procedure with statistical guarantees

We now provide a plug-in estimator for the optimal fair classifier g^*_{fair}. This algorithm enjoys strong theoretical guarantees in terms of both fairness and risk. In particular, Section 4.3.2 exhibits distribution-

free exact fairness guarantees.

4.3.1 Plug-in estimator

The enhanced estimation procedure is in two-steps. We first build estimators of the conditional probabilities $(p_k)_k$. Then, the estimation of parameters λ^* and $(\pi_s)_{s\in\mathcal{S}}$ is considered.

More precisely, our data-driven procedure is semi-supervised as it relies on two independent datasets, one labeled and another unlabeled. The first *labeled* dataset $\mathcal{D}_n = (X_i, S_i, Y_i)_{i=1,\dots,n}$ contains *i.i.d.* samples from the distribution $\mathbb{P}$. It allows to train estimators $(\hat{p}_k)_k$ of the conditional probabilities $(p_k)_k$, *e.g.,* Random Forest, SVM, *etc.* The second *unlabeled* dataset $\mathcal{D}'_N$ contains N *i.i.d.* copies of (X,S). It is used to calibrate fairness at the right level and estimate in particular several quantities such as marginal distributions. Therefore, $\mathcal{D}'_N$ is split in the following way: the *i.i.d.* sample $(S_1,\dots,S_N)$ of sensitive features is used to compute empirical frequencies $(\hat{\pi}_s)_{s\in\mathcal{S}}$ as estimates of $(\pi_s)_{s\in\mathcal{S}}$ (recall that $\pi_s = \mathbb{P}(S=s)$). For $s\in\mathcal{S}$, the number of observations corresponding to $S=s$ is denoted N_s, so that $N_{-1}+N_1 = N$. The feature vectors in $\mathcal{D}'_N$ are denoted $X^s_1,\dots,X^s_{N_s}$ and consist of *i.i.d.* data from the distribution $\mathbb{P}_{X^s}$ of $X|S=s$. All samples are assumed independent.

Remark 4.2

Classical datasets often only contain labeled samples. Then, our approach requires to split the data into two independent samples $\mathcal{D}_n$ and $\mathcal{D}'_N$, by removing labels in the latter. As illustrated in Appendix D, this splitting step is important to calibrate the right level of fairness.

We now discuss an important aspect of our procedure. Once the empirical conditional probabilities $\hat{p}_k(\cdot,\cdot)$ are trained, the theoretical analysis of the risk and the unfairness of the plug-in rule requires continuity conditions on the random variables $\hat{p}_k(X,S)$ (conditional on the learning sample, see Assumption 4.1). However, such property is automatically satisfied whenever perturbing $(\hat{p}_k)_k$ with a 'small' random noise. To be more specific, we introduce $\bar{p}_k(X,S,\zeta_k) := \hat{p}_k(X,S)+\zeta_k$, for a given uniform perturbation ζ_k on $[0,u]$. This perturbation improves the fairness calibration in both theory and practice. Without the perturbation, atoms may appear for the random variables $\hat{p}_k(X,S)-\hat{p}_j(X,S)$ and then no guarantee on the fairness (nor on the risk) can be established. On the other hand, its introduction does not deflate our theoretical study. In particular, our analysis, as in Theorem 4.2, takes the additional perturbation into account.

Let $(\zeta_k)_{k\in[K]}$ and $(\zeta^s_{k,i})$ be independent copies of a Uniform distribution on $[0,u]$. Because of this extra randomness, we call our fair algorithm $\hat{g}$ *randomized exactly fair classifier* and define it by plug-in as

$$\hat{g}(x,s) = \arg\max_{k\in[K]} \left(\hat{\pi}_s \bar{p}_k(x,s,\zeta_k) - s\hat{\lambda}_k\right), \tag{4.2}$$

for all $(x,s) \in \mathscr{X} \times \mathscr{S}$, with $\hat{\lambda} \in \mathbb{R}^K$ given as

$$\hat{\lambda} \in \arg\min_{\lambda} \sum_{s\in\mathscr{S}} \frac{1}{N_s} \sum_{i=1}^{N_s} \left[\max_{k\in[K]} \left(\hat{\pi}_s \bar{p}_k(X_i^s, s, \zeta_{k,i}^s) - s\lambda_k \right) \right]. \tag{4.3}$$

Note that the construction of the plug-in rule $\hat{g}$ relies on (x,s) but also on the perturbations ζ_k and $\zeta_{k,i}^s$ for $k \in [K]$, $i \in N_s$ and $s \in \mathscr{S}$. This additional data points are easy to collect since they are *i.i.d.* uniform random variables.

4.3.2 Statistical guarantees

We are now in position to derive fairness and consistency guarantees of our plug-in procedure.

Universal exact fairness guarantee. We first focus on fairness assessment and prove that the plug-in estimator $\hat{g}$ is asymptotically exactly fair. The convergence rate on the unfairness to zero is parametric with the number of unlabeled data N. Notably, the fairness guarantee is distribution-free and holds for any estimators of the conditional probabilities.

Theorem 4.1

There exists a constant $C > 0$ depending only on K and $\min_{s\in\mathscr{S}} \pi_s$, such that, for any estimators $\hat{p}_k$, we have

$$\mathbb{E}[\mathscr{U}(\hat{g})] \leq \frac{C}{\sqrt{N}} \quad .$$

This result illustrates a key feature of our post-processing approach. It makes (asymptotically) exactly fair any off-the-shelf (unconstrained) estimators of the conditional probabilities. This post-processing step is especially appealing when the cost of re-training an existing learning algorithm is high.

Consistency result. We now provide the consistency of $\hat{g}$ *w.r.t.* the misclassification risk. We define the L_1-norm in $\mathbb{R}^K$ between the estimator $\hat{\mathbf{p}} := (\hat{p}_1, \ldots, \hat{p}_K)$ and the vector of the conditional probabilities $\mathbf{p} := (p_1, \ldots, p_K)$ by $\|\hat{\mathbf{p}} - \mathbf{p}\|_1 = \sum_{k\in[K]} |\hat{p}_k(X,S) - p_k(X,S)|$.

Theorem 4.2

Let Assumption 4.1 be satisfied, then,

$$\mathbb{E}[\mathscr{R}_{\lambda^*}(\hat{g})] - \mathscr{R}_{\lambda^*}(g^*_{fair}) \leq C \left(\mathbb{E}\left[\|\hat{\mathbf{p}} - \mathbf{p}\|_1 \right] + \sum_{s\in\mathscr{S}} \mathbb{E}[|\hat{\pi}_s - \pi_s|] + \mathbb{E}[\mathscr{U}(\hat{g})] + u \right) \quad .$$

The above result highlights that the excess fair-risk of $\hat{g}$ depends on 1) the quality of the estimators of the conditional probabilities through its L_1-risk; 2) the quality of the estimators of $(\pi_s)_{s\in\mathscr{S}}$; 3) the

unfairness of the classifier; and 4) the upper-bound u on the regularizing perturbations. In view of Theorem 4.1, $\hat{g}$ is then consistent *w.r.t.* the misclassification risk as soon as the estimator $\hat{\mathbf{p}}$ is consistent in L_1-norm.

Corollary 2

If $\mathbb{E}\left[\|\hat{\mathbf{p}} - \mathbf{p}\|_1\right] \to 0$ and $u = u_n \to 0$ when $n \to \infty$, we have

$$|\mathbb{E}[\mathscr{R}(\hat{g})] - \mathscr{R}(g^*_{fair})| \to 0, \qquad \text{as} \quad n, N \to \infty \ .$$

We emphasize that Theorem 4.1 and Corollary 2 directly imply that $\hat{g}$ performs asymptotically as well as g^*_{fair} in terms of both fairness and accuracy: under suitable conditions, we have $\mathbb{E}[\mathscr{R}(\hat{g})] \to \mathscr{R}(g^*_{fair})$ and $\mathbb{E}[\mathscr{U}(\hat{g})] \to 0$ as $n \to \infty$.

Remark 4.3

The estimation of $\mathbf{p}$ is an important aspect of the procedure in order to get the consistency of $\hat{g}$. In Appendix C, we differ the study of an example of consistent learning algorithm based on ERM for which we derive a rate of convergence for the excess fair-risk in Theorem 4.2.

4.4 Approximate fair multi-class classification

Approximate fairness, also called ε-fairness, is particularly popular from a practical perspective in the field of algorithmic fairness. Importantly, the user is allowed to relax the fairness constraint whenever relevant or needed. Such relaxation is crucial when strict fairness strongly deflates the accuracy of the method. Of course, such modularity has a cost: the solution can not be as fair as the exact fair one; Besides, the unfairness level becomes a parameter that has no clear interpretation. Without clear justification, some empirical rules exist such as the forth-firth that tolerates an unfairness of 0.2 [HH00, Col07, FFM+15].

In this section we extend the results of Sections 4.2-4.3 in the *approximate* fairness setting, without taking in consideration the issue of selection of the level ε of unfairness.

4.4.1 ε-demographic parity in multi-class setting

First, we extend Definition 4.1 to the context of ε-fairness.

Definition 4.3: ε-Demographic parity (DP)

Let $\varepsilon \geq 0$, we say that a classifier $g \in \mathscr{G}$ is ε-fair (and write $g \in \mathscr{G}_{\varepsilon-fair}$) *w.r.t.* the distribution $\mathbb{P}$ on

$\mathcal{X} \times \mathcal{S} \times [K]$ if for each $k \in [K]$

$$|\mathbb{P}(g(X,S) = k|S = 1) - \mathbb{P}(g(X,S) = k|S = -1)| \leq \varepsilon \ .$$

When $\varepsilon = 0$, Definition 4.1 reduces to exact fairness. Analogously to the exact fairness setting, we need a formalism of ε fairness. To this end, we use again the measure of the unfairness $\mathcal{U}(\cdot)$ introduced in Definition 4.2.

Definition 4.4: ε-fairness

A classifier g is ε-fair if and only if $\mathcal{U}(g) \leq \varepsilon$.

4.4.2 Optimal fair classifier

Our goal is to derive an explicit formulation of the optimal ε-fair classifiers *w.r.t.* the misclassification risk, denoted by $g^*_{\varepsilon-fair}$, which is solution of

$$\min_{g \in \mathcal{G}_{\varepsilon-fair}} \mathcal{R}(g) \ .$$

Solving this problem shares similarities with the exact fairness case. However, deriving the optimal ε-fair classifier is trickier and requires different tools. Nevertheless, the first step remains to write the Lagrangian of the above problem: for $\lambda^{(1)} = (\lambda^{(1)}_1, \ldots, \lambda^{(1)}_K) \in \mathbb{R}^K_+$ and $\lambda^{(2)} = (\lambda^{(2)}_1, \ldots, \lambda^{(2)}_K) \in \mathbb{R}^K_+$ we define the *ε-fair-risk* as

$$\begin{aligned} \mathcal{R}_{\lambda^{(1)},\lambda^{(2)}}(g) := \mathcal{R}(g) &+ \sum_{k=1}^{K} \lambda^{(1)}_k [\mathbb{P}(g(X,S) = k|S = 1) - \mathbb{P}(g(X,S) = k|S = -1) - \varepsilon] \\ &+ \sum_{k=1}^{K} \lambda^{(2)}_k [\mathbb{P}(g(X,S) = k|S = -1) - \mathbb{P}(g(X,S) = k|S = 1) - \varepsilon] \ . \end{aligned} \tag{4.4}$$

An analog of Proposition 4.1 follows as well as a complete characterization of the optimal ε-fair classifier.

Proposition 4.2

Let $H : \mathbb{R}^{2K}_+ \to \mathbb{R}$ be the function

$$H(\lambda^{(1)}, \lambda^{(2)}) \quad = \quad \sum_{s \in \mathcal{S}} \mathbb{E}_{X|S=s} \left[\max_k \left(\pi_s p_k(X,s) - s(\lambda^{(1)}_k - \lambda^{(2)}_k) \right) \right] \ + \ \varepsilon \sum_{k=1}^{K} (\lambda^{(1)}_k \ + \ \lambda^{(2)}_k) \ .$$

Let Assumption 4.1 be satisfied and define $\lambda^{*(1)}, \lambda^{*(2)} \in \mathbb{R}^{2K}_+$ by

$$(\lambda^{*(1)}, \lambda^{*(2)}) \in \arg \min_{(\lambda^{(1)},\lambda^{(2)}) \in \mathbb{R}^{2K}_+} H(\lambda^{(1)}, \lambda^{(2)}) \ .$$

Then, $g^*_{\varepsilon-fair} \in \arg\min_{g \in \mathcal{G}_{\varepsilon-fair}} \mathcal{R}(g)$ if and only if $g^*_{\varepsilon-fair} \in \arg\min_{g \in \mathcal{G}} \mathcal{R}_{\lambda^{*(1)},\lambda^{*(2)}}(g)$.

In addition, for all $(x,s) \in \mathscr{X} \times \mathscr{S}$, we can rewrite

$$g^*_{\varepsilon-fair}(x,s) = \arg\max_{k\in[K]} \Big(\pi_s p_k(x,s) - s(\lambda_k^{*(1)} - \lambda_k^{*(2)}) \Big).$$

4.4.3 Plug-in ε fair classifier.

From now on, we strictly follow the methodology considered in the exact fairness setting. In particular, we derive an empirical counterpart of the classifier $g^*_{\varepsilon-fair}$ in a semi-supervised manner using the same datasets as in Section 4.3.1. The rule remains the same: estimate all unknown quantities and plug them into the expression of $g^*_{\varepsilon-fair}$. This allows to write the plug-in estimator

$$\hat{g}_\varepsilon(x,s) = \arg\max_{k\in[K]} \Big(\hat{\pi}_s \bar{p}_k(x,s,\zeta_k) - s(\hat{\lambda}_k^{(1)} - \hat{\lambda}_k^{(2)}) \Big) \quad , \tag{4.5}$$

for all $(x,s) \in \mathscr{X} \times \mathscr{S}$, where the couple $(\hat{\lambda}^{(1)}, \hat{\lambda}^{(2)})$ is minimizer over $\mathbb{R}_+^{2K}$ of $\hat{H}(\lambda^{(1)}, \lambda^{(2)})$ which is defined as

$$\hat{H}(\lambda^{(1)}, \lambda^{(2)}) = \sum_{s\in\mathscr{S}} \frac{1}{N_s} \sum_{i=1}^{N_s} \Big[\max_k \Big(\hat{\pi}_s \bar{p}_k(X_i^s, s, \zeta_{k,i}^s) - s(\lambda_k^{(1)} - \lambda_k^{(2)}) \Big) \Big] + \varepsilon \sum_{k=1}^{K} (\lambda_k^{(1)} + \lambda_k^{(2)}) \ . \tag{4.6}$$

Notice that here again, we exploited the randomization trick that is still required to set the correct level of fairness. We conclude this section by a remark regarding the fairness and the risk of the ε-fair estimator.

Going through the proofs of Theorems 4.1 and 4.2, we notice that they can be adapted to the ε fairness setting *modulo* minor changes. The only part that must be taken with care concerns the sub-differential of the empirical objective function that brings into play an additional restriction on the parameter space, since the dual variables in the Lagrangian are positive. This being said, we can extend Theorems 4.1 and 4.2 to the ε setting case and show that:

i) Distribution-free ε-fairness. For any estimators $\hat{p}_k$, the estimator $\hat{g}_\varepsilon$ achieves the right fairness level, that is, $|\mathbb{E}[\mathscr{U}(\hat{g}_\varepsilon)] - \varepsilon| \leq \frac{C}{\sqrt{N}}$ for some positive constant C.

ii) Consistency results. If the preliminary estimator of the conditional probabilities is consistent in L_1-norm, that is, if $\mathbb{E}\left[\|\hat{\mathbf{p}} - \mathbf{p}\|_1\right] \to 0$ and if $u = u_n \to 0$ when $n \to \infty$, then $\mathbb{E}[\mathscr{R}(\hat{g}_\varepsilon)] \to \mathscr{R}(g^*_{\varepsilon-fair})$ as $n, N \to \infty$.

This provides strong theoretical guarantees for $\hat{g}_\varepsilon$ in terms of both fairness and risk. Our approach can specifically control the fairness of the algorithm and set it at a desired level.

4.5 Implementation of the algorithm

In the present section, we focus on the implementation of our algorithm that produces an ε-fairness classifier. While the implementation in the exact fairness setting might be improved using accelerated

gradient descent, we do not develop it here and simply identify the exact fair algorithm as the approximate fair one whenever $\varepsilon = 0$.

The proposed approximate fair algorithm is defined in Eq. (4.5) and requires to solve an optimization problem in Eq. (4.6). In this section, we elaborate on the implementation–pseudo-code provided in Algorithm 5.

Algorithm 5 ε-fairness calibration

Input: ε parameter enabling the exact or approximate fairness, new data point (x, s), base estimators $(\bar{p}_k)_k$, unlabeled sample $\mathscr{D}'_N$, $(\zeta_k)_k$ and $(\zeta^s_{k,i})_{k,i,s}$ collection of i.i.d uniform perturbations in $[0, 10^{-5}]$

Step 0. Split $\mathscr{D}'_N$ and construct the samples $(S_1, \ldots, S_N)$ and $\{X^s_1, \ldots, X^s_{N_s}\}$, for $s \in \mathscr{S}$;
Step 1. Compute the empirical frequencies $(\hat{\pi}_s)_s$ based on $(S_1, \ldots, S_N)$;
Step 2. Compute $\hat{\lambda}^{(1)} = (\hat{\lambda}^{(1)}_1, \ldots, \hat{\lambda}^{(1)}_K)$ and $\hat{\lambda}^{(2)} = (\hat{\lambda}^{(2)}_1, \ldots, \hat{\lambda}^{(2)}_K)$ as a solution of Eq. (4.6);
`Sequential quadratic programming of Section 4.5 can be used for this step.`
Step 3. Compute $\hat{g}$ thanks to Eq. (4.5);
Output: ε-fair classification $\hat{g}(x, s)$ at point (x, s).

First of all, base estimators $(\bar{p}_k)_k$ are needed as input of the algorithm. In our numerical study, we consider Random Forest (RF), SVM, and logistic regression (reglog). However, we emphasize that for this step we can fit any off-the-shelf estimators by using the labeled dataset $\mathscr{D}_n$. In particular, already pre-trained efficient machine learning algorithms, whose retraining might be costly can be used. This is one of the main advantages of post-processing approaches as compared to in-processing ones. One might not forget the randomization in the definition of $\bar{p}_k$ that offers good theoretical properties for fairness calibration (see Section 4.3.1).

Once we have computed the $(\bar{p}_k)_k$, the fair classifier $\hat{g}$ relies on the estimators $\hat{\lambda}^{(1)}$ and $\hat{\lambda}^{(2)}$ computed in Step 2. of the algorithm. It requires solving the minimization problem given by Eq. (4.6). The corresponding objective function is convex but non-smooth due to the evaluation of the function *max* function. One classical way to regularize the objective function is to simply replace the hard-max by a soft-max Namely, for β a positive real number designating the temperature parameter and $x \in \mathbb{R}^K$, we set

$$\text{softmax}(x) := \sum_{k=1}^{K} \sigma_\beta(x)_k \cdot x_k \quad ,$$
$$\text{where} \quad \sigma_\beta(x)_k := \frac{\exp\left(x_k/\beta\right)}{\sum_{k=1}^{K} \exp\left(x_k/\beta\right)} \quad .$$

Whenever $\beta \to 0$ the soft-max reduces to the classical max function. Problem (4.6) with the soft-max relaxation is regular enough to be solved by a constrained optimization method, such as sequential quadratic programming [FLG19, Nie07]. Empirical study shows that $\beta = 0.005$ enables a good accuracy of our algorithm, without deviating too much from the original solution (with the max function).

Instead of regularizing the objective function, one can alternatively use sampling methods such as cross-entropy optimization [Rub99] on the original objective function. Despite their precision, the

downside of these algorithms is their computational complexity, whose growth with the problem dimension is much faster than the one of their smooth counterpart. For this reason, the regularization approach has been preferred in the following numerical study.

4.6 Numerical Evaluation

We now evaluate our method numerically[1]. Section 4.6.1 illustrates the efficiency of our ε-fairness algorithm on synthetic data, while experiments on various real datasets are provided in Section 4.6.2. Since, up to our knowledge, imposing fairness constraint in multi-class classification in a model-agnostic manner is not addressed in the literature we compare our method to the state-of-the-art approach proposed in [ADW19] for binary classification.

4.6.1 Evaluation on synthetic data

Synthetic data. Let define the synthetic data (X,S,Y). Conditional on $Y=k$ with $k \in [K]$, features $X \in \mathbf{R}^d$ follows a Gaussian mixture of m components, while the sensitive feature $S \in \{-1,+1\}$ follows a Bernoulli *contamination* with parameter p and $p-1$ if $k \leq \lfloor K/2 \rfloor$ and $k > \lfloor K/2 \rfloor$ respectively:

$$\begin{aligned}
(X|Y=k) &\sim \frac{1}{m}\sum_{i=1}^{m} \mathcal{N}_d(c^k+\mu_i^k, I_d), \quad \text{for } k \in [K], \\
(S|Y=k) &\sim 2\cdot\mathcal{B}(p)-1, \qquad \text{if} \quad k \leq \lfloor K/2 \rfloor, \\
(S|Y=k) &\sim 2\cdot\mathcal{B}(1-p)-1, \qquad \text{if} \quad k > \lfloor K/2 \rfloor,
\end{aligned}$$

with $c^k \sim \mathcal{U}_d(-1,1)$, and $\mu_1^k,\ldots,\mu_m^k \sim \mathcal{N}_d(0,I_d)$. Notably, this synthetic data structure enables to challenge different aspects of the algorithm. The parameter p measures the historical bias in the dataset. Specifically, the model becomes fair when $p=0.5$ and completely unfair when $p \in \{0,1\}$ (see Figure 4.1 for an illustration). As default parameters, we set $K=6$, $p=0.75$, $m=10$ and $d=20$.

Simulation scheme. We compare our method to the unfair approach. We set $u=10^{-5}$ and the probabilities p_k are estimated by RF with default parameters in `scikit-learn`. For all experiments, we generate $n=5000$ synthetic examples and split the data into three sets (60% training set, 20% hold-out set and 20% unlabelled set). The performance of a classifier g is evaluated by its empirical accuracy $Acc(g)$ on the hold-out set. The fairness of g is measured on the hold-out set via the empirical counterpart of the unfairness measure $\mathcal{U}(g)$ given in Definition 4.2. We repeat each procedure 30 times in order to report the average performance (accuracy and unfairness) alongside its standard deviation on the hold-out set.

Fairness versus Accuracy. Fig. 4.2 illustrates the evolution of the performance (unfairness and accuracy) of the algorithm with respect to ε. Fig. 4.3-Left displays the fairness and accuracy of our

[1]The source of our method can be found at `https://github.com/curiousML/epsilon-fairness`.

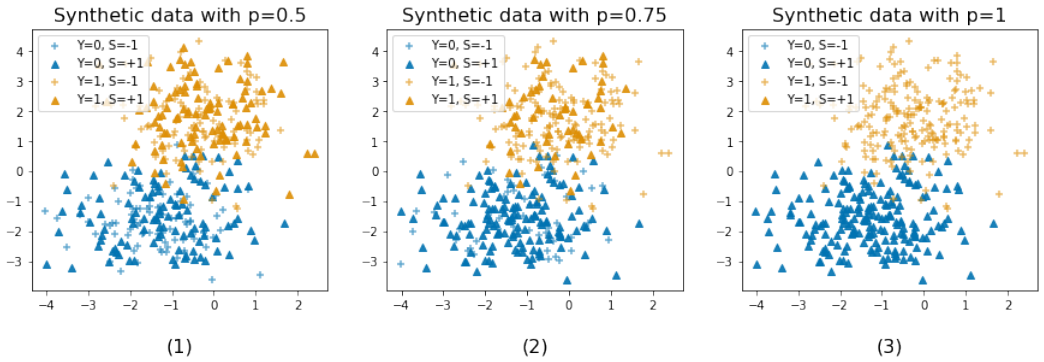


Figure 4.1: Example of synthetic data in binary case.
We set $d = 2$ and $m = 1$. The level of unfairness is set as follows: *(1)* $p = 0.5$ (*e.g.* no unfairness); *(2)* $p = 0.75$ (*e.g.* unfair dataset); *(3)* $p = 1$ (*e.g.* highly unfair dataset).

algorithm for different levels of historical bias (quantified by p) in the dataset. The evolution of the performance in Fig. 4.2 behaves as expected: enforcing more fairness is counter-balanced by a weaker accuracy (see also in Fig. 4.3), therefore the trade-off between unfairness and accuracy can be controlled by the parameter ε. In particular, in case of exact fairness $\varepsilon = 0$, the gain in fairness is particularly salient and effective. By contrast, whenever $\varepsilon = 0.15$, the fair classifier becomes similar to the unfair method, meaning that the original unfairness of the problem is around $\varepsilon = 0.15$. From Fig. 4.3-Left, we additionally notice that: 1) the fairness efficiency of the algorithm is particularly significant for datasets with large historical bias ($p = 0.95$ or 0.99); 2) our method succeeds to reach the demanded unfairness level up to small approximation terms (see how the curves are vertical as soon as the bound on the unfairness is reached).

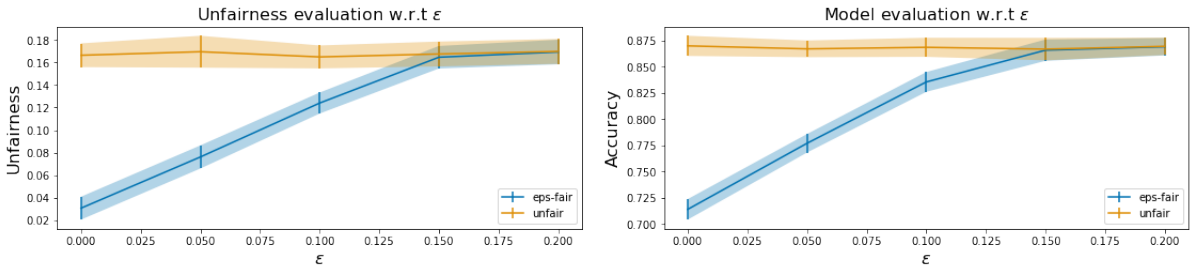


Figure 4.2: Performance of ε-fair and unfair classifiers in terms of accuracy and fairness.
Left: evolution of the unfairness *w.r.t.* ε; *Right*: evolution of the accuracy *w.r.t.* ε.
We report the means and standard deviations over 30 repetitions.

Fairness at the level of scores. Fig. 4.4 confirms our findings in Proposition 1: ε-fairness is enforced by shifting the conditional probabilities (*e.g.*, the exact fairness with $\varepsilon = 0$). When ε moves away from 0 (approximate fairness), the conditional probabilities translate a situation where the ε-fair classifier

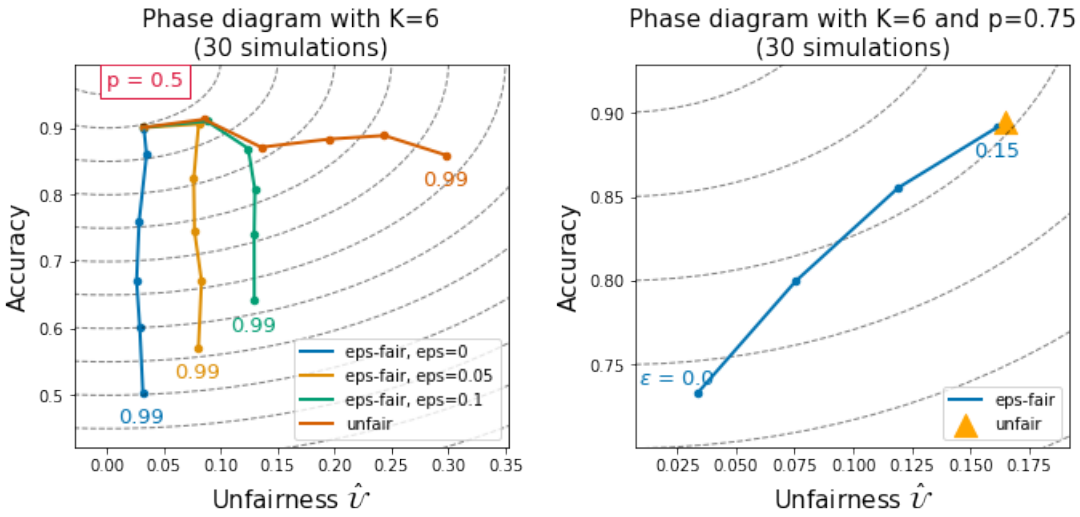


Figure 4.3: (Accuracy, Unfairness) phase diagrams for synthetic datasets.
(Accuracy, Unfairness) phase diagrams *w.r.t. Left* the level of bias p; *Right* the accuracy-fairness trade-off parameter ε . Top-left corner gives the best trade-off.

becomes more unfair. Here again we observe, but at the score distributions level, the matching between the unfair and the ε-fair classifiers whenever $\varepsilon = 0.15$.

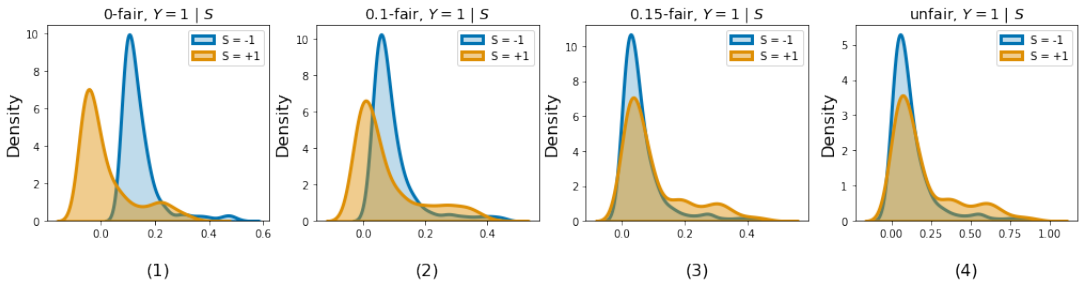


Figure 4.4: Empirical distribution of the score functions for the class $Y = 1$, conditional to the sensitive feature $S = \pm 1$.
(1)-(3) ϵ-fairness with $\epsilon \in \{0, 0.1, 0.15\}$, *(4)* unfair.

4.6.2 Application to real datasets

Methods. We compare our ε-fair method for both linear and non-linear multi-class classification. For linear models, we consider the one-versus-all logistic regression (reglog); for non-linear models, we use SVM model with Gaussian kernel (gaussSVC) and RF. Hyperparemeters are provided in Appx. D. Note that in multi-class setting, the accuracy of a random guess for a balanced dataset is around $1/K$.

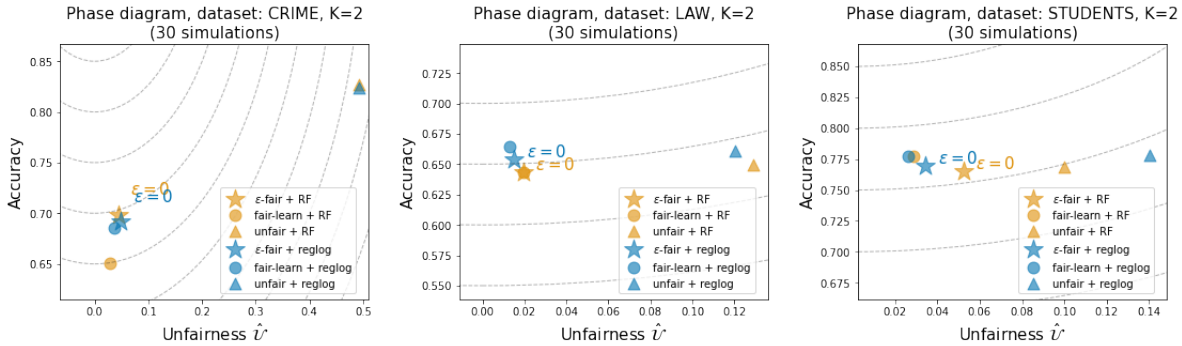


Figure 4.5: (Accuracy, Unfairness) phase diagrams that shows the performance of the methods. Top-left corner gives the best trade-off.

Datasets. The performance of our method is evaluated on three benchmark datasets : CRIME, LAW and STUDENTS Hereafter, we provide a short description of these datasets.

- *Communities&Crime* (CRIME) dataset contains socio-economic, law enforcement, and crime data about communities in the US with 1994 examples. The task is to predict the number of violent crimes per 10^5 population which, we divide into $K = 5$ balanced classes based on equidistant quantiles. Following [CKK+13] the binary sensitive feature is the percentage of black population.

- *Law School Admissions* (LAW) dataset [WR98] presents national longitudinal bar passage data and has 20649 examples. The task is to predict a students GPA divided into $K = 3$ classes based on equidistant quantiles. The sensitive attribute is the race (white versus non-white).

- *Student Performance* (STUDENTS) dataset [CS08] is about student achievement in two Portuguese high schools. The task is to predict the number of grades passed (out of 3 grades *i.e.*, $K = 4$) based on student social and school related features. In binary case we predict whether the student passed the final grade. The sensitive attribute is the student age (≥ 18 vs. < 18).

Performance in binary case ($K = 2$). Before considering the multi-class setting, we analyze the relevancy of our proposal for binary classification.

In this context, the state-of-the-art is established by the in-processing approach in [ADW19] that penalizes unfairness and will serve as baseline[2]. We focus the comparison between our approach and the state-of-the-art benchmark one to $\varepsilon = 0$ (the exact fairness problem) and illustrate in Fig 4.5 the performance of the methods on the LAW, CRIME and STUDENTS. While common belief suggests that in-processing methods outperform post-processing methods, it appears that our post-processing exact fairness approach is very efficient both in terms of accuracy and fairness. Indeed, the numerical experiments reveal that our method is competitive in several aspects: 1) Competitive unfairness.

[2]The method in [ADW19] was developed for *Equality of Odds* but their code is also implemented for *Demographic Parity* see https://github.com/fairlearn/fairlearn

Overall, our exactly-fair algorithm achieves similar performance as the state-of-the-art benchmark one, when we consider reglog or RF. 2) Competitive accuracy. While on LAW and STUDENTS we achieve approximately the same accuracy, we achieve a better accuracy on CRIME when we consider RF (0.70 vs 0.65). 3) Time complexity. Since the method proposed in [ADW19] is an in-processing algorithm, using the dedicated package, the running time of the baseline is much more higher than with our method.

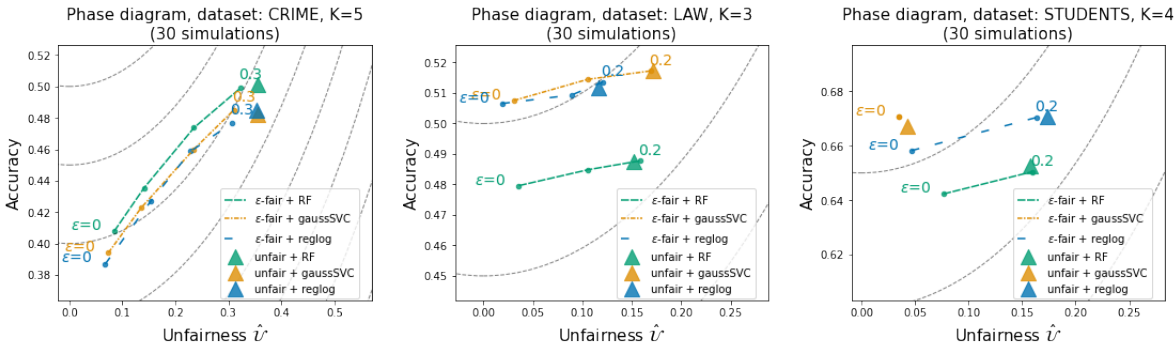


Figure 4.6: (Accuracy, Unfairness) phase diagrams in binary case.
(Accuracy, Unfairness) phase diagrams that shows the evolution, *w.r.t.* the accuracy fairness trade-off parameter ε. Top-left corner gives the best trade-off.

Performance in multi-class case ($K \geq 3$). Numerical experiments in the multi-class setting are presented in Fig. 4.6. They confirm our observations on synthetic data. The performance of the ε-fairness algorithm gets closer to the unfair method as we relax the constraint on fairness. In addition, the results highlight the effectiveness of our method in enforcing fairness as ε decreases. In particular, the fairness calibration is close to the pre-specified level, regardless of the base algorithm (reglog, gaussSVC, or RF).

4.7 Conclusion

In the multi-class classification framework, we provide an optimal fair classification rule under DP constraint and derive misclassification and fairness guarantees of the associated plug-in fair classifier (see Alg. 5). We handle both exact and approximate fairness settings and show that our approach achieves distribution-free fairness and can be applied on top of any probabilistic base estimator. We illustrate the proficiency of our procedure on various synthetic and real datasets. In particular, our algorithm is efficient for enforcing a pre-specified level of fairness.

Calibrating the level of unfairness $\varepsilon \geq 0$ might be desired in some situations. A future direction of research is to describe a methodology that statistically justifies a data-driven calibration of this parameter in order to optimally compromise risk and unfairness.

Appendices

In this section, we gather the proof of our results. In all the sequel, C denotes a generic constant, whose value may vary from line to line.

A Proof for exact fairness

This section is devoted to the proof of the results given in Section 4.2 and 4.3.

A.1 Proof for fair optimal rule

We begin with an auxiliary lemma, which provides an alternative useful representation of $\mathscr{R}_\lambda(g)$.

Lemma 1

The ε-fair-risk of a classifier g with balancing parameter $\lambda = (\lambda_1, \dots, \lambda_K) \in \mathbb{R}^K$ rewrites:

$$\mathscr{R}_\lambda(g) = \sum_{s \in \mathscr{S}} \mathbb{E}_{X|S=s} \left[\sum_{k=1}^{K} (\pi_s p_k(X,S) - s\lambda_k) \mathbb{1}_{\{g(X,S) \neq k\}} \right] . \tag{7}$$

Proof of Lemma 1 Let $\lambda \in \mathbb{R}^K$ and recall the following definition of the fair-risk

$$\begin{aligned} \mathscr{R}_\lambda(g) =& \mathbb{P}(g(X,S) \neq Y) \\ & - \sum_{s \in \mathscr{S}} \sum_{k=1}^{K} s\lambda_k \mathbb{P}_{X|S=s}(g(X,S) \neq k) \ . \end{aligned}$$

We have the following decomposition

$$\begin{aligned} \mathbb{P}(g(X,S) \neq Y) &= \sum_{k=1}^{K} \mathbb{E}\left[\mathbb{1}_{\{g(X,S) \neq k\}} \mathbb{1}_{\{Y=k\}} \right] \\ &= \sum_{k=1}^{K} \sum_{s \in \mathscr{S}} \mathbb{E}\left[\mathbb{1}_{\{g(X,S) \neq k\}} \mathbb{1}_{\{S=s\}} p_k(X,S) \right] \\ &= \sum_{k=1}^{K} \sum_{s \in \mathscr{S}} \mathbb{E}_{X|S=s}\left[\mathbb{1}_{\{g(X,s) \neq k\}} \pi_s p_k(X,s) \right] \ , \end{aligned}$$

which directly implies (7). ■

Proof of Proposition 4.1 Recall that g^*_λ minimizes $\mathscr{R}_\lambda$ on $\mathscr{G}$. Besides, we deduce from Lemma 1 that

$$\mathscr{R}_\lambda(g^*_\lambda) = 1 - \sum_{s \in \mathscr{S}} \mathbb{E}_{X|S=s} \left[\max_{k \in [K]} (\pi_s p_k(X,s) - s\lambda_k) \right] . \tag{8}$$

Hence, a maximizer λ^* in $\mathbb{R}^K$ of $\lambda \mapsto \mathscr{R}_\lambda(g^*_\lambda)$ takes the form

$$\lambda^* \in \arg\min_{\lambda \in \mathbb{R}^K} \sum_{s \in \mathscr{S}} \mathbb{E}_{X|S=s} \left[\max_{k \in [K]} (\pi_s p_k(X,s) - s\lambda_k) \right] .$$

The above criterion is convex in λ. Therefore, first order optimality conditions for the minimization over λ of the above criterion imply that, for each $k \in [K]$,

$$0 = \sum_{s\in\mathcal{S}} s\mathbb{P}_{X|S=s}\left(\forall j \neq k\ (\pi_s p_k(X,s) - s\lambda_k^*) > (\pi_s p_j(X,s) - s\lambda_j^*)\right)$$
$$+ su_k^s \mathbb{P}_{X|S=s}\left(\forall j \neq k\ (\pi_s p_k(X,s) - s\lambda_k^*) \geq (\pi_s p_j(X,s) - s\lambda_j^*),\ \exists j \neq k\ (\pi_s p_k(X,s) - s\lambda_k^*) = (\pi_s p_j(X,s) - s\lambda_j^*)\right) \ ,$$

with $u_k^s \in [0,1]$ for all $k \in [K]$ and $s \in \mathcal{S}$. Thanks to Assumption 4.1, $p_k(X,s) - p_j(X,s)$ has no atoms for all $s \in \mathcal{S}$ and then the second part of the r.h.s. vanishes. Therefore for all $k \in [K]$

$$\mathbb{P}_{X|S=1}\left(g^*_{\lambda^*}(X,S) \neq k\right) = \mathbb{P}_{X|S=-1}\left(g^*_{\lambda^*}(X,S) \neq k\right) \ ,$$

meaning that the classifier $g^*_{\lambda^*}$ is fair. Furthermore, for any fair classifier $g \in \mathcal{G}_{fair}$, we observe that

$$\mathcal{R}(g^*_{\lambda^*}) = \mathcal{R}_{\lambda^*}(g^*_{\lambda^*}) \leq \mathcal{R}_{\lambda^*}(g) = \mathcal{R}(g),$$

so that $g^*_{\lambda^*}$ is also an optimal fair classifier.

Conversely, consider any optimal fair classifier $g^*_{fair} \in \mathcal{G}_{f\dashv\rangle\nabla}$. Combining the fairness of g^*_{fair} with the optimality of λ^* over the family $(R_\lambda(g^*_\lambda))_{\lambda\in\mathbf{R}^K}$, we deduce

$$\mathcal{R}_{\lambda^*}(g^*_{fair}) = \mathcal{R}(g^*_{fair}) \leq \mathcal{R}_{\lambda^*}(g^*_{\lambda^*}) \leq \mathcal{R}_{\lambda^*}(g), \text{for any } g \in \mathcal{G}\,.$$

Hence any optimal fair classifier is a minimizer of $\mathcal{R}_{\lambda^*}$ over $\mathcal{G}$. ■

Proof of Corollary 1 The proof follows directly from Lemma 1 and Proposition 4.1. In particular, Eq. (8) implies that

$$g^*_\lambda \in \arg\min_g \mathcal{R}_{\lambda^*}(g),$$

is characterized by

$$g^*_{\lambda^*}(x,s) \in \arg\max_{k\in[K]} \left(\pi_s p_k(x,s) - s\lambda_k^*\right).$$

■

A.2 Proof of Consistency results

We start this section with two results, Lemmas 2-3 that directly follow from similar arguments as in the proofs of Proposition A.2. and Lemma B.8 in [CDH+20b] respectively. Their proofs are hence omitted.

Lemma 2

The parameter λ^*, and $\hat{\lambda}$ are bounded.

Lemma 3

We have that, for each $s \in \mathscr{S}$ and $k \in [K]$,

$$\mathbb{E}\left[\frac{1}{N_s}\sum_{i=1}^{N_s} \mathbb{1}\{\exists j \neq k, \hat{h}_k^s(X_i, \hat{\lambda}_k) = \hat{h}_j^s(X_i, \hat{\lambda}_j)\}\right] \leq \frac{C}{N_s} \ ,$$

where $\hat{h}_k^s : (x, \lambda) \mapsto \hat{\pi}_s \bar{p}_k(x, s) - s\lambda$.

Let us now consider the proofs of Theorem 4.1 and Theorem 4.2.

Proof of Theorem 4.1 As in Lemma 3, we first introduce, for $s \in \mathscr{S}$ and $k \in [K]$,

$$\hat{h}_k^s : (x, \lambda) \mapsto \hat{\pi}_s \bar{p}_k(x, s) - s\lambda \ .$$

By construction, the estimator $\bar{p}_k(X, S)$ satisfies Assumption 4.1, therefore for all $s \in \mathscr{S}$ and $k \in [K]$

$$\mathbb{P}_{X|S=s}(\hat{g}(X, S) = k) = \mathbb{P}_{X|S=s}\left(\forall j \neq k, \hat{h}_k^s(X, \hat{\lambda}_k) > \hat{h}_j^s(X, \hat{\lambda}_j)\right) \ .$$

Now, let us make use of the optimality of $\hat{\lambda}$. We denote by $\hat{\mathbb{P}}_{X|S=s}$ the empirical measure on the data $\{X_1^s, \ldots, X_{N_s}^s\}$. Considering the first order optimality conditions for $\hat{\lambda}$, we can show that, for all $k \in [K]$ and $s \in \mathscr{S}$, there exists $\alpha_k^s \in [-1, 1]$ such that

$$s\hat{\mathbb{P}}_{X|S=s}\left(\forall j \neq k, \hat{h}_k^s(X, \hat{\lambda}_k) > \hat{h}_j^s(X, \hat{\lambda}_j)\right) + \\ \alpha_k^s \hat{\mathbb{P}}_{X|S=s}\left(\forall j \neq k, \hat{h}_k^s(X, \hat{\lambda}_k) \geq \hat{h}_j^s(X, \hat{\lambda}_j),\ \exists j \neq k, \hat{h}_k^s(X, \hat{\lambda}_k) = \hat{h}_j^s(X, \hat{\lambda}_j)\right) = 0 \ .$$

From the above equation, we deduce that

$$\begin{aligned}\mathscr{U}(\hat{g}) = \sum_{k=1}^{K} &\left|\mathbb{P}_{X|S=1}(\hat{g}(X, S) = k) - \mathbb{P}_{X|S=-1}(\hat{g}(X, S) = k)\right| \\ &\leq \sum_{k=1}^{K}\sum_{s\in\mathscr{S}} \left|\left(\mathbb{P}_{X|S=s} - \hat{\mathbb{P}}_{X|S=s}\right)\left(\forall j \neq k,\ \hat{h}_k^s(X, \hat{\lambda}_k) > \hat{h}_j^s(X, \hat{\lambda}_j)\right)\right| \\ &\quad + \sum_{k=1}^{K}\sum_{s\in\mathscr{S}} \hat{\mathbb{P}}_{X|S=s}\left(\exists j \neq k,\ \hat{h}_k^s(X, \hat{\lambda}_k) = \hat{h}_j^s(X, \hat{\lambda}_j)\right) \ .\end{aligned}$$

Observe that for all $k \in [K]$

$$\begin{aligned}&\left|\left(\mathbb{P}_{X|S=s} - \hat{\mathbb{P}}_{X|S=s}\right)\left(\forall j \neq k,\ \hat{h}_k^s(X, \hat{\lambda}_k) > \hat{h}_j^s(X, \hat{\lambda}_j)\right)\right| = \\ &\qquad \left|\left(\mathbb{P}_{X|S=s} - \hat{\mathbb{P}}_{X|S=s}\right)\left(\forall j \neq k,\ \bar{p}_k(X, s) - \bar{p}_j(X, s) \geq \frac{s(\hat{\lambda}_k - \hat{\lambda}_j)}{\hat{\pi}_s}\right)\right| \\ &\qquad\qquad \leq \sum_{j=1}^{K} \sup_{t\in\mathbb{R}} \left|\left(\mathbb{P}_{X|S=s} - \hat{\mathbb{P}}_{X|S=s}\right)\left(\bar{p}_k(X, s) - \bar{p}_j(X, s) \geq t\right)\right| \ .\end{aligned}$$

Therefore, from the Dvoretzky-Kiefer-Wolfowitz Inequality conditional on $\mathscr{D}_n$ and on $(S_1,\ldots,S_N)$, we deduce that, for each $s \in \mathscr{S}$ and $k \in [K]$

$$\mathbb{E}\left[\left|\left(\mathbb{P}_{X|S=s} - \hat{\mathbb{P}}_{X|S=s}\right)\left(\forall j \neq k, \hat{h}_k^s(X,\hat{\lambda}_k) > \hat{h}_j^s(X,\hat{\lambda}_j)\right)\right|\right] \leq C\sqrt{\frac{1}{N_s}} \quad .$$

Applying Lemma 3, we then get that, Conditional on $\mathscr{D}_n$ and on $(S_1,\ldots,S_N)$, we have that

$$\mathbb{E}[\mathscr{U}(\hat{g})] \leq C\sum_{s\in\mathscr{S}}\sqrt{\frac{1}{N_s}} \quad .$$

Since N_s is a binomial random variable with parameter (π_s, N), we get

$$\mathbb{E}[\mathscr{U}(\hat{g})] \leq C\sqrt{\frac{1}{N}},$$

where C depends on K and $\min(\pi_{-1},\pi_1)$. ■

Proof of Theorem 4.2 The proof goes conditional on the training data. First, let us decompose *excess fair-risk* of the classifier $\hat{g}$ in a convenient way for our analysis

$$\mathscr{R}_{\lambda^*}(\hat{g}) - \mathscr{R}_{\lambda^*}(g^*_{fair}) = \left(\mathscr{R}_{\lambda^*}(\hat{g}) - \mathscr{R}_{\hat{\lambda}}(\hat{g})\right) + \left(\mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\lambda^*}\left(g^*_{\lambda^*}\right)\right) \quad , \tag{9}$$

where we recall that $g^*_{fair} = g^*_{\lambda^*}$. We propose to deal with the two terms in r.h.s. of Equation (9) separately. According the the first term, we have

$$\begin{aligned}\left(\mathscr{R}_{\lambda^*}(\hat{g}) - \mathscr{R}_{\hat{\lambda}}(\hat{g})\right) &= \sum_{s\in\mathscr{S}}\sum_{k=1}^{K} s\lambda^*_k \mathbb{P}_{X|S=s}(\hat{g}(X,S)\neq k) - \sum_{s\in\mathscr{S}}\sum_{k=1}^{K} s\hat{\lambda}_k \mathbb{P}_{X|S=s}(\hat{g}(X,S)\neq k) \\ &= \sum_{s\in\mathscr{S}}\sum_{k=1}^{K} s\left(\lambda^*_k - \hat{\lambda}_k\right)\mathbb{P}_{X|S=s}(\hat{g}(X,S)\neq k) \quad .\end{aligned}$$

Since, for each $k \in [K]$, the parameters λ^*_k and $\hat{\lambda}_k$ are bounded (see Lemma 2), we deduce that

$$\mathscr{R}_{\lambda^*}(\hat{g}) - \mathscr{R}_{\hat{\lambda}}(\hat{g}) \leq C\mathscr{U}(\hat{g}) \quad . \tag{10}$$

Then we have shown that the first term in the r.h.s. of Eq. (9) relies on the unfairness of the classifier $\hat{g}$. Now, let us consider the second term in r.h.s. of Equation (9). Our goal will be to show that this term mainly depends on the quality of the base estimators $\hat{p}_k$. Since λ^* is a maximizer of $\mathscr{R}_\lambda(g^*_\lambda)$ over λ, it is clear that, conditional on the data, $\mathscr{R}_{\lambda^*}(g^*_{\lambda^*}) \geq \mathscr{R}_{\hat{\lambda}}(g^*_{\hat{\lambda}})$. (The parameter $\hat{\lambda}$ is seen as fixed conditional on the data.) Therefore, we have

$$\mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\lambda^*}\left(g^*_{\lambda^*}\right) \leq \mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\hat{\lambda}}(g^*_{\hat{\lambda}}) \quad .$$

By introducing $\hat{g}^*_{\hat{\lambda}}$, we remove the estimation of λ^* from the study of $\mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\lambda^*}\left(g^*_{\lambda^*}\right)$. At this point, it becomes clear that bounding this term does not relies on the unlabeled sample sizes N_s. Let us recall the definition of $g^*_{\hat{\lambda}}$: conditional on the data

$$g^*_{\hat{\lambda}} \in \arg\min_{g\in\mathscr{G}} \mathscr{R}_{\hat{\lambda}}(g) \quad .$$

Then using similar arguments as those leading to Eq. (8) implies that (see also Corollary 1)

$$g^*_{\hat{\lambda}} \in \arg\max_{k\in[K]} \left(\pi_s p_k(x,s) - s\hat{\lambda}_k\right) \quad .$$

As a consequence, using the writing of the fair-risk provided by Lemma 1

$$\mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\hat{\lambda}}(g^*_{\hat{\lambda}}) = \sum_{s\in\mathscr{S}} \mathbb{E}_{X|S=s}\left[\max_{k\in[K]}\left(\pi_s p_k(X,S) - s\hat{\lambda}_k\right) - \sum_{k=1}^{K}\left(\pi_s p_k(X,S) - s\hat{\lambda}_k\right)\mathbb{1}_{\{\hat{g}(X,S)=k\}}\right] \quad .$$

Because of the indicator function, there is only one non-zero element in the inner sum. Then we observe that for each $s \in \mathscr{S}$

$$\begin{aligned}\left|\max_{k\in[K]}\left(\pi_s p_k(X,S) - s\hat{\lambda}_k\right) - \sum_{k=1}^{K}\left(\pi_s p_k(X,S) - s\hat{\lambda}_k\right)\mathbb{1}_{\{\hat{g}(X,S)=k\}}\right| \\ \leq 2\max_{k\in[K]}\left|\left(\pi_s p_k(X,S) - s\hat{\lambda}_k\right) - \left(\hat{\pi}_s \bar{p}_k(X,S) - s\hat{\lambda}_k\right)\right| \\ \leq 2\left(\max_{k\in[K]}|p_k(X,S) - \bar{p}_k(X,S)| + |\pi_s - \bar{\pi}_s|\right) \quad ,\end{aligned}$$

where the last inequality is due to the fact that $\pi_s, \hat{\pi}_s, p_k$, and $\bar{p}_k$ are all in $[0,1]$. Therefore, recalling that $\bar{p}_k$ is a randomized version of $\hat{p}_k$ we can write

$$\mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\hat{\lambda}}(g^*_{\hat{\lambda}}) \leq C\left(\|\hat{\mathbf{p}} - \mathbf{p}\|_1 + \sum_{s\in\mathscr{S}}|\hat{\pi}_s - \pi_s| + u\right) \quad ,$$

and obtain the bound

$$\mathscr{R}_{\hat{\lambda}}(\hat{g}) - \mathscr{R}_{\lambda^*}\left(g^*_{\lambda^*}\right) \leq C\left(\|\hat{\mathbf{p}} - \mathbf{p}\|_1 + \sum_{s\in\mathscr{S}}|\hat{\pi}_s - \pi_s| + u\right) \quad .$$

In view of Equation (9), the above inequality together with Equation (10) yield the desired result. ■

B Proof for approximate fairness

We begin with an auxiliary lemma, which provides an alternative useful representation of $\mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g)$.

Lemma 4

The fair-risk of a classifier g with balancing parameters $\lambda^{(1)} = (\lambda^{(1)}_1, \ldots, \lambda^{(1)}_K) \in \mathbb{R}^K_+, \lambda^{(2)} = (\lambda^{(2)}_1, \ldots, \lambda^{(2)}_K) \in \mathbb{R}^K_+$ reads as:

$$(11)\quad \mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g) = \sum_{s\in\mathscr{S}} \mathbb{E}_{X|S=s}\left[\sum_{k=1}^{K}\left(\pi_s p_k(X,S) - s(\lambda^{(1)}_k - \lambda^{(2)}_k)\right)\mathbb{1}_{\{g(X,S)\neq k\}}\right] - \varepsilon\sum_{k=1}^{K}(\lambda^{(1)}_k + \lambda^{(2)}_k).$$

Proof of Lemma 4 Let $(\lambda^{(1)}, \lambda^{(2)}) \in \mathbb{R}^{2K}_+$ and recall the following definition of the ε-fair-risk

$$\mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g) = \mathbb{P}(g(X,S) \neq Y) - \sum_{k=1}^{K}\sum_{s\in\mathscr{S}} s(\lambda^{(1)}_k - \lambda^{(2)}_k)\mathbb{E}_{X|S=s}\left[\mathbb{1}_{\{g(X,s)\neq k\}}\right] - \varepsilon\sum_{k=1}^{K}(\lambda^{(1)}_k + \lambda^{(2)}_k)$$

The result in (11) directly follows from the following decomposition

$$\begin{aligned}\mathbb{P}(g(X,S)\neq Y) &= \sum_{k=1}^{K}\mathbb{E}\left[\mathbb{1}_{\{g(X,S)\neq k\}}\mathbb{1}_{\{Y=k\}}\right]\\ &= \sum_{k=1}^{K}\sum_{s\in\mathscr{S}}\mathbb{E}\left[\mathbb{1}_{\{g(X,S)\neq k\}}\mathbb{1}_{\{S=s\}}p_k(X,S)\right]\\ &= \sum_{k=1}^{K}\sum_{s\in\mathscr{S}}\mathbb{E}_{X|S=s}\left[\mathbb{1}_{\{g(X,s)\neq k\}}\pi_s p_k(X,s)\right] \ .\end{aligned}$$

■

Proof of Proposition 4.2 From Lemma 4, we deduce that $g^*_{\lambda^{(1)},\lambda^{(2)}}$ should be defined for all $(x,s)\in\mathscr{X}\times\mathscr{S}$ as

$$g^*_{\lambda^{(1)},\lambda^{(1)}}(x,s) = \arg\max_{k\in[K]}\left(\pi_s p_k(X,S) - s(\lambda_k^{(1)}-\lambda_k^{(1)})\right) \ ,$$

since it minimizes the risk $\mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}$. Now we should maximize $\mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g^*_{\lambda^{(1)},\lambda^{(2)}})$ in the dual variables. Notice that the ε-fair risk can be wirten as

$$\mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g^*_{\lambda^{(1)},\lambda^{(2)}}) = 1 - \sum_{s\in\mathscr{S}}\mathbb{E}_{X|S=s}\left[\max_{k\in[K]}\left(\pi_s p_k(X,S) - s(\lambda_k^{(1)}-\lambda_k^{(2)})\right)\right] - \varepsilon\sum_{k=1}^{K}(\lambda_k^{(1)}+\lambda_k^{(2)}) \ .$$

Hence, a maximizer $(\lambda^{*(1)},\lambda^{*(2)})$ in $\mathbb{R}_+^{2K}$ of $(\lambda^{(1)},\lambda^{(2)})\mapsto\mathscr{R}_{\lambda^{(1)},\lambda^{(2)}}(g^*_{\lambda^{(1)},\lambda^{(2)}})$ takes the form

$$(\lambda^{*(1)},\lambda^{*(2)})\in\arg\min_{(\lambda^{(1)},\lambda^{(2)})\in\mathbb{R}_+^{2K}}\underbrace{\sum_{s\in\mathscr{S}}\mathbb{E}_{X|S=s}\left[\max_{k\in[K]}\left(\pi_s p_k(X,s) - s(\lambda_k^{(1)}-\lambda_k^{(2)})\right)\right] + \varepsilon\sum_{k=1}^{K}(\lambda_k^{(1)}+\lambda_k^{(2)})}_{H(\lambda^{(1)},\lambda^{(2)})} \ .$$

The rest of the proof consists in showing that such a calibration of the tuning parameters $\lambda^{(1)},\lambda^{(2)}$ implies that $g^*_{\lambda^{(1)},\lambda^{(2)}}$ is indeed ε-fair. Observe that

$$H(\lambda^{(1)},\lambda^{(2)}) \geq \varepsilon\sum_{k=1}^{K}(\lambda_k^{(1)}+\lambda_k^{(2)}) \ ,$$

and then $\lim_{\|(\lambda^{(1)},\lambda^{(2)})\|_2^2\to\infty}H(\lambda^{(1)},\lambda^{(2)}) = +\infty$. Moreover, this criterion is convex in $(\lambda^{(1)},\lambda^{(2)})$. Therefore the minimum $(\lambda^{*(1)},\lambda^{*(2)})$ exists. In particular, we can derive the first order optimality conditions of the above problem *w.r.t.* $\lambda^{(1)}$ which implies that for each $k\in[K]$,

$$\begin{aligned}0 = &-\sum_{s\in\mathscr{S}}s\mathbb{P}_{X|S=s}\left(\forall j\neq k\ (\pi_s p_k(X,s) - s(\lambda_k^{*(1)}-\lambda_k^{*(2)})) > (\pi_s p_j(X,s) - s(\lambda_j^{*(1)}-\lambda_j^{*(2)}))\right)\\ &+ s\,u_k^s\,\mathbb{P}_{X|S=s}\Big(\forall j\neq k\ (\pi_s p_k(X,s) - s(\lambda_k^{*(1)}-\lambda_k^{*(2)})) \geq (\pi_s p_j(X,s) - s(\lambda_j^{*(1)}-\lambda_j^{*(2)})),\\ &\qquad\exists j\neq k\ (\pi_s p_k(X,s) - s(\lambda_k^{*(1)}-\lambda_k^{*(2)})) = (\pi_s p_j(X,s) - s(\lambda_j^{*(1)}-\lambda_j^{*(2)}))\Big) + \varepsilon \ ,\end{aligned}$$

with $u_k^s \in [0,1]$ for all $k \in [K]$ and all $s \in \mathscr{S}$. Similarly the first order conditions *w.r.t.* $\lambda^{(1)}$ implies that for each $k \in [K]$,

$$0 = \sum_{s\in\mathscr{S}} s\mathbb{P}_{X|S=s}\Big(\forall j \neq k\ (\pi_s p_k(X,s) - s(\lambda_k^{*(1)} - \lambda_k^{*(2)})) > (\pi_s p_j(X,s) - s(\lambda_j^{*(1)} - \lambda_j^{*(2)}))\Big)$$
$$+ s\,v_k^s\,\mathbb{P}_{X|S=s}\Big(\forall j \neq k\ (\pi_s p_k(X,s) - s(\lambda_k^{*(1)} - \lambda_k^{*(2)})) \geq (\pi_s p_j(X,s) - s(\lambda_j^{*(1)} - \lambda_j^{*(2)})),$$
$$\exists j \neq k\ (\pi_s p_k(X,s) - s(\lambda_k^{*(1)} - \lambda_k^{*(2)})) = (\pi_s p_j(X,s) - s(\lambda_j^{*(1)} - \lambda_j^{*(2)}))\Big) + \varepsilon \ ,$$

with $v_k^s \in [0,1]$ for all $k \in [K]$ and $s \in \mathscr{S}$. Thanks to Assumption 4.1, $p_k(X,s) - p_j(X,s)$ has no atom for all $s \in \mathscr{S}$ and then the second part of the r.h.s. of both above equations vanish. Hence, the first order optimality conditions *w.r.t.* $\lambda_k^{*(1)}$ becomes

$$\mathbb{P}_{X|S=1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right) - \mathbb{P}_{X|S=-1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right) = \varepsilon \ ,$$

and the first order optimality conditions *w.r.t.* $\lambda_k^{*(2)}$ writes as

$$\mathbb{P}_{X|S=1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right) - \mathbb{P}_{X|S=-1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right) = -\varepsilon \ .$$

Therefore, we deduce the following constraint on $\lambda_k^{*(1)}, \lambda_k^{*(2)}$

$$\lambda_k^{*(1)}\lambda_k^{*(2)} = 0 \ and \ \lambda_k^{*(1)} + \lambda_k^{*(2)} \geq 0 \ .$$

Hence, if $\lambda_k^{*(1)} + \lambda_k^{*(2)} > 0$,

$$\left|\mathbb{P}_{X|S=1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right) - \mathbb{P}_{X|S=-1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right)\right| = \varepsilon.$$

In the case where $\lambda_k^{*(1)} = \lambda_k^{*(2)} = 0$, we use Euler Inequality and deduce

$$\left|\mathbb{P}_{X|S=1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right) - \mathbb{P}_{X|S=-1}\left(g^*_{\lambda^{*(1)},\lambda^{*(2)}}(X,S) \neq k\right)\right| \leq \varepsilon \ .$$

■

C Rates of convergence for ERM estimator

We have established in Section 4.3.2 theoretical guarantees on risk and fairness for our plug-in procedure, when using any off-the-shelf consistent estimator of the conditional probability. In this section, we study more close detail the classical setting where the conditional probabilities estimation step is provided by ERM and derive an explicit bound on the fair-risk of the resulting fair classifier,

For a given (measurable) score function $f(\cdot) = (f_1(\cdot), \ldots, f_K(\cdot))$ mapping $\mathscr{X} \times \{-1,1\}$ onto $\mathbb{R}^K$, we define the induced classification rule [3]

$$g_f(\cdot) \in \arg\max_{k\in[K]} f_k(\cdot) \ .$$

[3] Whenever the maximum is reached at multiple indices, we set by convention g_f as the smallest index in $[K]$.

For the sake of simplicity, we focus on the L_2-risk[4]

$$R_2(f) := \mathbb{E}\left[\sum_{k=1}^{K}(Z_k - f_k(X,S))^2\right] \ ,$$

where $Z_k := 2\mathbb{1}_{\{Y=k\}} - 1$.

The optimal score function f^* *w.r.t.* R_2 is denoted $f^* := \arg\min_f R_2(f)$, where the infimum is taken over all measurable functions that map $\mathscr{X} \times \{-1,1\}$ onto $\mathbb{R}^K$. The optimum f^* satisfies the relation $f_k^*(X,S) = 2p_k(X,S) - 1$. In particular, Zhang's Lemma [Zha04] implies that $\mathbb{E}\left[\mathscr{R}(g_f) - \mathscr{R}(g^*)\right] \leq (\mathbb{E}[R_2(f) - R_2(f^*)])^{1/2}$, for any score function f. This inequality highlights the connection between the usual misclassification risk of g_f and the L_2-risk of the score function f.

Let us now consider the empirical counterpart of the L_2-risk R_2, given for any function f by

$$\hat{R}_2(f) := \frac{1}{n}\sum_{i=1}^{n}\sum_{k=1}^{K}\left(Z_k^i - f_k(X_i,S_i)\right)^2,$$

with $Z_k^i := 2\mathbb{1}_{\{Y_i=k\}} - 1$. The empirical risk minimizer $\hat{f}$ over a given convex set $\mathscr{F}$ of functions is given by

$$\hat{f} \in \arg\min_{f\in\mathscr{F}} \hat{R}_2(f) \ .$$

In view of the expression for the optimal score function f^*, we naturally set $\hat{p}_k := \frac{\hat{f}_k+1}{2}$ as an estimator of the conditional probability p_k. Given $\hat{p}(\cdot) = (\hat{p}_1(\cdot), \ldots, \hat{p}_K(\cdot))$, our plug-in procedure given in Eq. (4.2) provides an exactly fair classifier that we denote by $g_{\hat{f}}$. According to the theoretical study conducted in Section 4.3.2, Theorem 4.1 ensures that the classifier $g_{\hat{f}}$ is asymptotically exactly fair.

According to the analysis of the fair-risk of the classifier $g_{\hat{f}}$, we invoke Theorem 4.2 complemented with the consistency of the empirical risk minimizer $\hat{f}$ *w.r.t.* to the L_2-risk (due to Zhang's Lemma). In order to specify the precise rate of convergence of the fair-risk, we introduce additional assumptions on the class of functions $\mathscr{F}$.

Assumption 0.2

The set $\mathscr{F}$ satisfies the following:

1. There exists $B > 0$ s.t. $\|\mathbf{f}\|_\infty := \max_{k\in[K]} \sup_{x\in\mathscr{X}} |f_k(x)| \leq B$, for each $\mathbf{f} \in \mathscr{F}$;
2. For $\varepsilon > 0$, there exists an ε-net $\mathscr{F}_\varepsilon \subset \mathscr{F}$ *w.r.t.* $\|.\|_\infty$ and a positive constant $C_{\mathscr{F}}$ s.t. $\log(|\mathscr{F}_\varepsilon|) \leq C_{\mathscr{F}} \log(\varepsilon^{-1})$.

Note that Assumption 0.2 covers bounded parametric classes among others (for instance, linear or polynomial with bounded degree and bounded coefficients score functions). This structural assumption on the set of models $\mathscr{F}$ enables to control the rate of convergence of the excess fair-risk of $g_{\hat{f}}$ the constructed fair classifier.

[4]Since the $0-1$ loss lacks convexity, we consider the square loss as a convex surrogate to avoid computational issues.

Proposition 0.3

Assume that $f^* \in \mathscr{F}$ and $u = u_n = \frac{1}{n}$. If Assumptions 4.1 and 0.2 hold, then

$$\mathbb{E}\left[\mathscr{R}_{\lambda^*}(g_{\hat{f}}) - \mathscr{R}_{\lambda^*}(g^*_{fair})\right] \leq C\left(\left(\frac{\log(n)}{n}\right)^{1/2} + N^{-1/2}\right) .$$

Under classical assumptions on the complexity of $\mathscr{F}$, Proposition 0.3 induces in particular a parametric rate of convergence for the excess fair-risk of $g_{\hat{f}}$.

Proof. From Theorems 4.1 and 4.2, we have that

$$\mathbb{E}\left[\mathscr{R}_{\lambda^*}(g_{\hat{f}}) - \mathscr{R}_{\lambda^*}(g^*_{fair})\right] \quad \leq \quad \mathbb{E}\|\hat{\boldsymbol{p}} - \mathbf{p}\|_1 + \frac{1}{n} + \frac{C}{\sqrt{N}} .$$

Since

$$\mathbb{E}\|\hat{\boldsymbol{p}} - \mathbf{p}\|_1 \leq \frac{1}{2}\mathbb{E}\left\|\hat{\boldsymbol{f}} - \boldsymbol{f}^*\right\|_1 \leq \frac{1}{2}\sqrt{\sum_{k=1}^{K}\mathbb{E}\left[\left(f_k(X,S) - f_k^*(X,S)\right)^2\right]} ,$$

it remains to provide a control on the term $\sum_{k=1}^{K}\mathbb{E}\left[\left(f_k(X,S) - f_k^*(X,S)\right)^2\right]$. For this purpose, let us first prove that for each score function $\mathbf{f} \in \mathscr{F}$, the following holds

$$\sum_{k=1}^{K}\mathbb{E}\left[\left(f_k(X,S) - f_k^*(X,S)\right)^2\right] \leq 2\left(R_2(\mathbf{f}) - R_2(\mathbf{f}^*)\right) . \tag{12}$$

Indeed, we observe that

$$\frac{(Z_k - f_k)^2 + (Z_k - f_k^*)^2}{2} - \left(Z_k - \left(\frac{f_k + f_k^*}{2}\right)\right)^2 = \frac{\left(f_k - f_k^*\right)^2}{4} .$$

From this equality, we then deduce that

$$\frac{1}{4}\sum_{k=1}^{K}\mathbb{E}\left[\left(f_k(X,S) - f_k^*(X,S)\right)^2\right] = \frac{1}{2}\left(R_2(\mathbf{f}) + R_2(\mathbf{f}^*)\right) - R_2\left(\frac{\mathbf{f}+\mathbf{f}^*}{2}\right) .$$

Since R_2 is positive, we get Equation (12).

The next step of the proof is to bound $R_2(\hat{\mathbf{f}}) - R_2(\mathbf{f}^*)$. We have by definition of $\hat{\mathbf{f}}$

$$R_2(\hat{\mathbf{f}}) - R_2(\mathbf{f}^*) \leq R_2(\hat{\mathbf{f}}) - R_2(\mathbf{f}^*) - 2\left(\hat{R}_2(\hat{\mathbf{f}}) - \hat{R}_2(\mathbf{f}^*)\right).$$

Furthermore from Assumption 0.2 with $\varepsilon = 1/n$, we have

$$R_2(\hat{\mathbf{f}}) - R_2(\mathbf{f}^*) - 2\left(\hat{R}_2(\hat{\mathbf{f}}) - \hat{R}_2(\mathbf{f}^*)\right) \leq \frac{C}{n} + \sup_{f \in \mathscr{F}_{1/n}}\left\{R_2(\mathbf{f}) - R_2(\mathbf{f}^*)\right\} - 2\left(\hat{R}_2(\hat{\mathbf{f}}) - \hat{R}_2(\mathbf{f}^*)\right) . \tag{13}$$

If we denote $\mathrm{Err}(\mathbf{f}) = R_2(\mathbf{f}) - R_2(\mathbf{f}^*)$ and $\widehat{\mathrm{Err}}(\mathbf{f}) = \hat{R}_2(\hat{\mathbf{f}}) - \hat{R}_2(\mathbf{f}^*)$, from Bernstein's Inequality together with Assumption 0.2, we have, for all $t > 0$ and $\mathbf{f} \in \mathscr{F}_\varepsilon$,

$$\begin{aligned} & \mathbb{P}\left(\mathrm{Err}(\mathbf{f}) - 2 \cdot \widehat{\mathrm{Err}}(\mathbf{f}) \geq t\right) \\ \leq \; & \mathbb{P}\left(2\left(\mathrm{Err}(\mathbf{f}) - \widehat{\mathrm{Err}}(\mathbf{f})\right) \geq t + \mathrm{Err}(\mathbf{f})\right) \\ \leq \; & \exp\left(-\frac{\frac{n}{8} \cdot (t + \mathbb{E}[h(Z, \mathbf{f}(X,S))])^2}{\mathbb{E}[|h(Z, \mathbf{f}(X,S))|^2] + \frac{C}{3} \cdot (t + \mathbb{E}[h(Z, \mathbf{f}(X,S))])}\right) \end{aligned}$$

where $h(Z, \mathbf{f}(X,S)) := \sum_{k=1}^{K} (|Z - f_k(X,S)|^2 - |Z - f_k^*(X,S)|^2)$. Furthermore, observe that

$$\mathbb{E}[|h(Z, \mathbf{f}(X,S))|^2] \leq C \cdot \mathbb{E}[h(Z, \mathbf{f}(X,S))] \;\; ,$$

which, plugged in the previous inequality, directly provides

$$\mathbb{P}\left(\mathrm{Err}(\mathbf{f}) - 2 \cdot \widehat{\mathrm{Err}}(\mathbf{f}) \geq t\right) \quad \leq \quad \exp(-Cnt) \;\; .$$

Hence, combining a union bound argument together with Assumption 0.2 and (13), we compute

$$\mathbb{E}[R_2(\hat{\mathbf{f}}) - R_2(\mathbf{f}^*)] \leq \frac{C}{n} + CC_{\mathscr{F}} \frac{\log(n)}{n} \;\; . \tag{14}$$

Plugging this inequality in (12) concludes the proof. ■

D Numerical experiments

Hyperparameters. The hyperparamerers are set with default parameters in scikit-learn except the number of trees for RF which is set at 500.

Splitting the sample. Our theoretical study relies on the independence of the datasets $\mathscr{D}_n$ and $\mathscr{D}'_n$. Figure .7 illustrates the importance of such condition for the fairness but also the accuracy of our proposed *argmax-fair* method. Indeed, whenever splitting is not performed (left parts of plots), the fairness performance of the fair algorithm can even be worse than the unfair method. This emphasize that splitting is crucial and enables to avoid over-fitting on the training set.

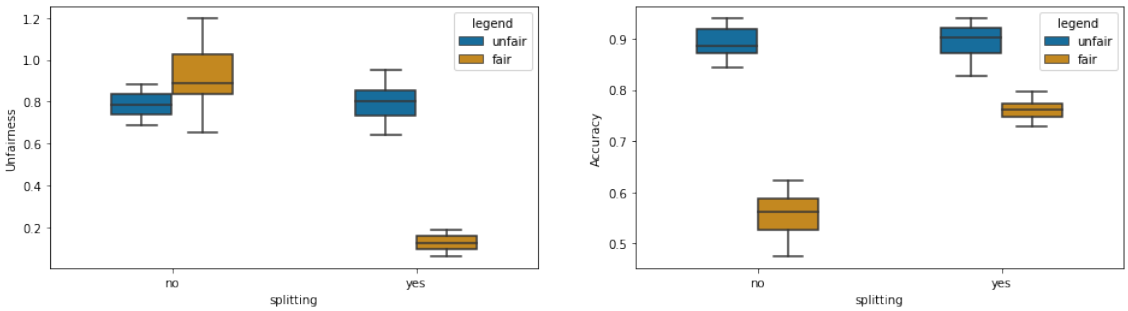


Figure .7: Empirical impact of data splitting.
Empirical impact of data splitting on unfairness (Left – the lower the better) and accuracy (Right: accuracy – the higher the better). Boxplots generated over 30 repetitions with $p = 0.75$. Left: unfairness – the lower the better; Right: accuracy – the higher the better.

Chapter 5

Fairness in multi-class classification : alternative method

Contents

Keywords: Algorithmic fairness, multi-class classification, statistical learning.

5.1 Introduction

In this chapter, we challenge several numerical aspects of the exact-fair classifier procedure presented in chapter 4. As a benchmark, we introduce an alternative approach that enforces fairness on each individual score which is a variant of [YX20]. Then, we illustrate the efficiency on our procedure to build fair reliable predictions on synthetic and real world datasets.

5.2 Benchmark alternative approach for fair multi-class classification : score-fair classifier

The procedure developed in Chap.4 enforces the score maximizer to be fair whereas [YX20] imposes fairness at each score function. We call them respectively *argmax-fair* classifier and *score-fair* classifier.

Note that in *argmax-fair* classifier we consider only the case of exact fairness with $\epsilon = 0$.

Definition 5.1

We say that $f : \mathscr{X} \times \{-1,1\} \mapsto \mathbb{R}^K$ is *score-fair* in demographic parity if each coordinate of f is fair *w.r.t.* the demographic parity notion of fairness.

Consequently, a possible way to tackle this problem is to consider the following minimization task

$$f^*_{score-fair} \in \operatorname{argmin} \{R_2(f) : f \text{ is } score\text{-}fair\} \quad . \tag{5.1}$$

where $R_2(f) = \mathbb{E}\left[\sum_{k=1}^K (Z_k - f_k(X,S))^2\right]$ and f is *score-fair* means that for all $k \in [K]$ and for all $t \in \mathbb{R}$ we have

$$\mathbb{P}(f_k(X,S) \leq t \mid S = -1) = \mathbb{P}(f_k(X,S) \leq t \mid S = 1) \quad .$$

The optimal solution for the problem (5.1) is separable and can then be solved element-wise. Thm. 2.3 in [CDH+20c] identifies the distribution of score-fair classifier $f^*_{score-fair}$ as solutions of a Wasserstein barycenter problem. In particular, [CDH+20c] allows us to deduce the explicit form $f^*_{score-fair} = (f^*_{sf,1}, \ldots, f^*_{sf,K}) \in \mathbb{R}^K$ such that

$$f^*_{sf,k}(x,s) = \left(\pi_{-s} Q_{f^*_k|-s}\right) \circ F_{f^*_k|s}\left(f^*_k(x,s)\right) \quad . \tag{5.2}$$

where for all $s \in \mathscr{S}$, $F_{f^*_k|s}$ is the Cumulative Distribution Function (CDF) of $f^*_k(X)|S = s$ and $Q_{f^*_k|s} : [0,1] \to \mathbb{R}$ is the corresponding quantile function defined for all $t \in (0,1]$ as $Q_{f^*_k|s}(t) = \inf\left\{y \in \mathbb{R} : F_{f^*_k|s}(y) \geq t\right\}$. Hence, it only remains to estimate each $f^*_{sf,k}$ by plug-in. More precisely, we need to estimate for all $s \in \mathscr{S}$, the proportion π_s, the CDF $F_{f^*_k|s}$ and the quantile function $Q_{f^*_k|s}$.

While this approach seem to be rather natural, let us emphasize that *score-fair* DP does not imply DP for the score maximizer, since the maximum, unlike thresholding, operation does not preserve the DP property. Optimal *score-fair* functions rely on the L_2-risk and be easily characterized following the approach in [GLR20, CDH+20c].

5.3 Pseudo-code for score-fair algorithm

Algorithm 6 proposes a pseudo-code for the implementation of an estimator $\hat{g}_{score-fair}$ of the classifier $g^*_{score-fair}$ deduced from the *score-fair classifier* given by

$$g^*_{score-fair}(x,s) = \arg\max_{k \in [K]} f^*_{sf,k}(x,s), \quad \forall (x,s) \in \mathscr{X} \times \mathscr{S} \quad .$$

We use in Algorithm 6 a close methodology to the one considered in Section 4.3. In particular, the base estimators $\bar{f}_k$ in the **Input** of the algorithm relies on an estimator $\hat{f}_k$ of the k-th element of the optimal score function $\mathbf{f}^*$ and on a uniform perturbation $(\zeta_k)_k$. Also, in **Step 0-a.** the constructed

Algorithm 6 ERM Score-fair classifier

Input: new data point (x,s), base estimators $(\bar{f}_k)_k$, unlabeled sample $\mathscr{D}'_N$,
$(\zeta_k)_k$ and $(\zeta^{j,s}_{k,i})_{k,i,j,s}$ collection of i.i.d uniform perturbations in $[0,10^{-5}]$
Step 0-a. Separate $\mathscr{D}'_N$ to construct two samples $(S_1,\dots,S_N)$ and $\mathscr{D}_{\mathscr{X}} = \{X_1,\dots,X_N\}$;
Step 0-b. Split $\mathscr{D}_{\mathscr{X}}$ into two samples $\mathscr{D}_{\mathscr{X},1}$ and $\mathscr{D}_{\mathscr{X},2}$ of size $N/2$;
Step 0-c. Split $\mathscr{D}_{\mathscr{X},j}$ into two samples $\mathscr{D}^s_{\mathscr{X},j} = \{X^s_{j,1},\dots,X^s_{j,N_s}\}$ with $s \in \mathscr{S}$;
Step 1. Compute the empirical frequencies $(\hat{\pi}_s)_s$ based on $(S_1,\dots,S_N)$;
for $k = 1$ **to** K **do**
For all $s \in \mathscr{S}$, estimate the CDF $F_{f^*_k|s}$ based on $\mathscr{D}^s_{\mathscr{X},1}$;
`Jittering with` $(\zeta^{1,s}_{k,i})_{k,i,1,s}$ `is needed for this step.`
For all $s \in \mathscr{S}$, estimate the quantile function $Q_{f^*_k|s}$ based on $\mathscr{D}^s_{\mathscr{X},2}$;
`Jittering with` $(\zeta^{2,s}_{k,i})_{k,i,2,s}$ `is needed for this step.`
Compute $\hat{f}_{sf,k}(x,s)$, the estimator of $f^*_{sf,k}(x,s)$ given in Eq. (5.2) by plug-in;
end for
Output: fair classifier $\hat{g}_{sf,k}(x,s) = \arg\max_{k\in[K]} \hat{f}_{sf,k}(x,s)$ at point (x,s).

sample $\mathscr{D}_{\mathscr{X}} = \{X_1,\dots,X_N\}$ consists only of the covariates from $\mathscr{D}'_N$. In **Step 0-b.** we split the sample $\mathscr{D}_{\mathscr{X}}$ into two sets $\mathscr{D}_{\mathscr{X},1}$ and $\mathscr{D}_{\mathscr{X},2}$ with size[1] $N/2$.

5.4 Evaluation on synthetic data

Synthetic data. As in Chap. 4, conditional on $Y = k$, the feature X comes from a Gaussian mixture, while the sensitive feature S follows a Bernoulli *contamination*. We recall that this synthetic dataset enables to challenge different aspects of our algorithm : The parameter p measures the historical bias in the dataset.

Simulation scheme. We compare our method to the benchmark *score-fair* alternative algorithm and the baseline unfair approach. We set $u = 10^{-5}$ and the probabilities p_k are estimated by RF with default parameters in `scikit-learn`. For all experiments, we generate $n = 600$ synthetic examples per class and we split the data into three sets (60% training set, 20% hold-out set and 20% unlabelled set). The performance of a classifier g is evaluated by its empirical accuracy $Acc(g)$ on the hold-out set. The fairness of g is measured on the hold-out set via the empirical counterpart of the unfairness measure $\mathscr{U}(g)$ given in Definition 4.2. We repeat this procedure 30 times in order to report the average performance (accuracy and unfairness) alongside its standard deviation on the hold-out set.

Fairness versus Accuracy. Fig. 5.1 displays the fairness and accuracy performances of our algorithm for different levels of historical bias in the dataset (measured by p). Our algorithm outperforms both score-fair and unfair classifier, in terms of fairness efficiency. However, such fairness performance

[1] For simplification of the presentation we assume that N is an even integer.

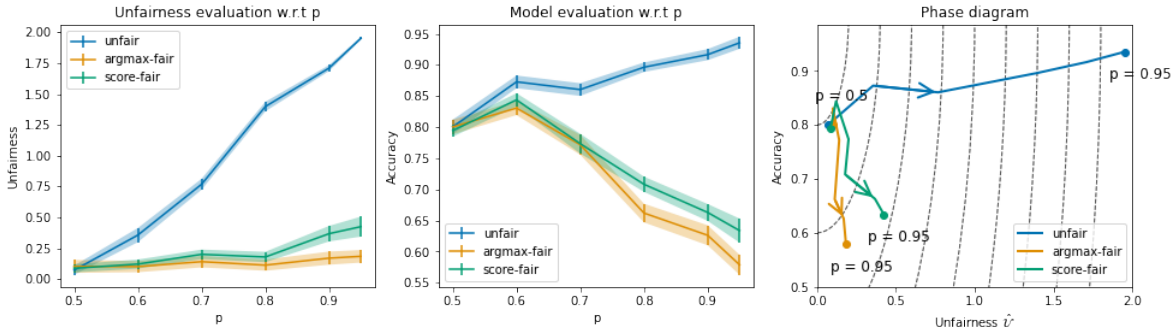


Figure 5.1: Performance of the classification procedures in terms of accuracy and fairness for the *unfair*, the *argmax-fair*, and the *score-fair* classifiers.

Left: evolution of the unfairness *w.r.t.* p; Middle: evolution of the accuracy *w.r.t.* p; Right: (Accuracy, Unfairness) phase diagram that shows the evolution, *w.r.t.* p, of trade-off between accuracy and fairness. The arrows go from fairest to most unfair situations. Top-left corner in the diagram gives the best trade-off.

is directly counter-balanced by a weaker accuracy, as visualised on the phase diagram (Unfairness, Accuracy) in Fig. 5.1-right. Fairness efficiency of our methods is particularly significant for datasets with large historical bias ($p = 0.9$ or 0.95).

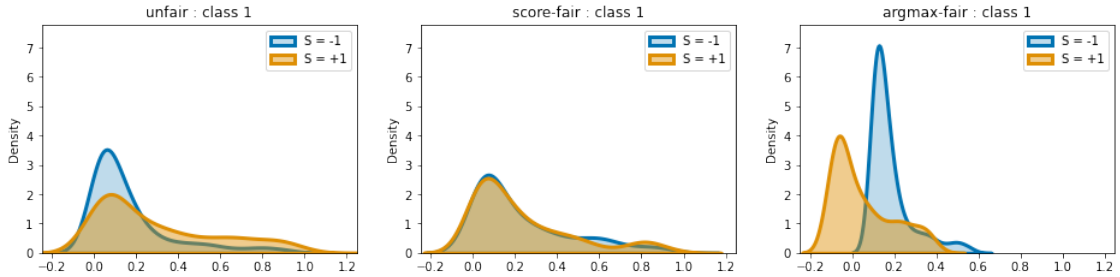


Figure 5.2: Empirical distribution of the score functions for the class $Y = 1$, conditional to the sensitive feature values $S = \pm 1$.

unfair (left), *score-fair* middle and *argmax-fair* (right) classifiers.

Fairness at the level of scores. Whereas both *argmax-fair* and *score-fair* approaches succeed to build fair classification, these two methods differ significantly on their impact on scores. Fig. 5.2 highlights this difference for the specific class $Y = 1$, but similar behavior for other classes. The right panel confirms our findings in Proposition 1: *argmax-fair* enforces fairness by shifting the conditional probabilities. Also expected is the fairness efficiency of *score-fair* (middle plot), while the resulting scores are easier to interpret: the distributions of the predictions for both sensitive features merge.

Additional remarks. Additional numerical illustrations display the effectiveness of our procedure on the synthetic data. We mainly *(i)* justify our choice of the temperature β in Figure 5.3 ; *(ii)* show the

effectiveness of both *score-fair* and *argmax-fair* classifier in terms of unfairness reduction in Figure 5.4, and *(iii)* illustrate our method's robustness with respect to the number of classes K in Figure 5.5. By default, as in the main body, we set the number of classes $K = 4$ in all our experiments.

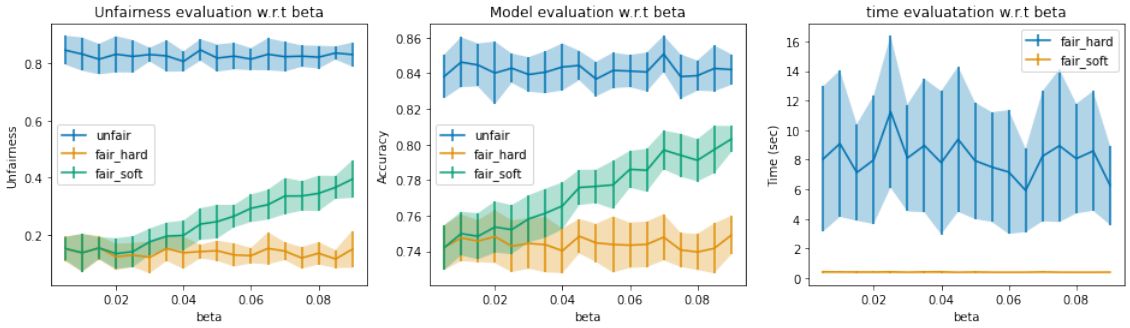


Figure 5.3: Performance of the classification procedures in terms of accuracy, fairness and time complexity.

Performance for the *fair_soft* (argmax-fair classifier with soft-max evaluation) classifier obtained from Algorithm 5 with the acceleration scheme in **Step 2**. The *unfair* and the *fair_hard* (*argmax-fair* classifier with hard-max evaluation [Rub99] – obtained by Algorithm 5 without acceleration in **Step 2.**) classifiers are used as baselines and Random Forest is used as base estimator. Left: evolution of the unfairness *w.r.t.* the temperature β; Middle: evolution of the accuracy *w.r.t.* β; Right: evolution of the time complexity *w.r.t.* β. We report the means and standard deviations over 30 simulations. The figure shows that both *fair_soft* and *fair_hard* have the comparable performance in terms of unfairness and accuracy for $\beta \leq 0.01$. However *fair_soft* is considered much faster than *fair_hard* hence *fair_soft* is chosen.

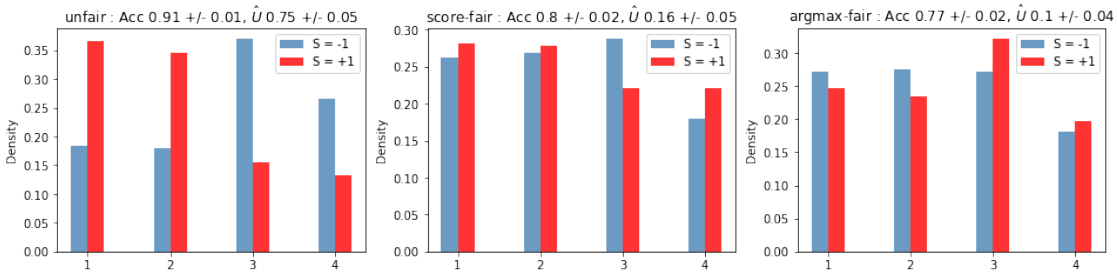


Figure 5.4: Empirical distribution of the *unfair* (left), the *score-fair* (middle) and the *argmax-fair* (right) classifiers conditional to the sensitive feature $S = \pm 1$

Each performance (accuracy and unfairness) is evaluated over 30 simulations and we consider RF as the base estimator. The histograms display the effectiveness of both *score-fair* and *argmax-fair* in enforcing fairness by rendering the empirical distributions across the two groups ($S = -1$ and $S = +1$) close. As shown by this empirical study, *argmax-fair* outperforms the *score-fair* classifier in terms of fairness.

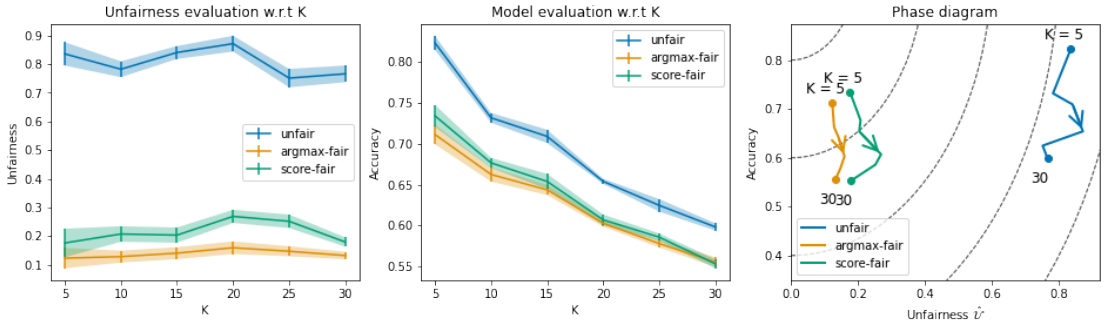


Figure 5.5: Performance of the classification procedures in terms of accuracy and unfairness for the *unfair*, the *argmax-fair*, and the *score-fair* classifiers.

RF is used as base estimator. Left: evolution of the unfairness *w.r.t.* the number of classes K; Middle: evolution of the accuracy *w.r.t.* K; Right: (Accuracy, Unfairness) phase diagram that shows the evolution that highlights a trade-off between accuracy and fairness *w.r.t.* K. The arrows go from fairest to most unfair situations. Top-left corner in the diagram gives the best trade-off. We report the means and standard deviations over 30 simulations. The figure shows that the increase in the number of classes doesn't impact the unfairness of each method and *argmax-fair* remains more effective in fairness than the other two methods.

5.5 Application to real datasets

Methods. We compare our method *argmax-fair* and the alternative approach *score-fair* for both linear and non-linear multi-class classification. For linear models, we consider the one-versus-all logistic regression (reglog) and the SVM with linear kernel (linearSVC); for non-linear models: SVM model with Gaussian kernel (GaussSVC) and RF. The hyperparamerers are set with default parameters in scikit-learn except the number of trees for RF which is set at 500.

Datasets. The performance of our method is evaluated on four benchmark datasets[2] : CRIME, LAW, WINE and CMC. Hereafter, we provide a short description of these datasets.

- *Communities&Crime* (CRIME) dataset contains socio-economic, law enforcement, and crime data about communities in the US with 1994 examples. The task is to predict the number of violent crimes per 10^5 population which, we divide into $K = 7$ balanced classes based on equidistant quantiles. Following [CKK+13] and [CDH+20a] the binary sensitive feature is the percentage of black population.

- *Law School Admissions* (LAW) dataset [WR98] presents national longitudinal bar passage data and has 20649 examples. The task is to predict a students GPA divided into $K = 4$ classes based on equidistant quantiles. The sensitive attribute is the race (white versus non-white).

[2]Some are used in Chap. 4

- *Wine Quality* (WINE) dataset [CCA+09] reports the description of 6497 wines and the task is to predict the quality graded by the experts. The quality is between 3 (bad) and 9 (good) but we consider only $K = 5$ classes (4 to 8) due to a too low frequency of the class 3 and 9 (resp. 5 and 30 examples). The sensitive attribute is the color (red versus white).

- *Contraceptive Method Choice* (CMC) dataset is about 1987 National Indonesia Contraceptive Prevalence Survey. The problem is to predict the contraceptive method choice of a woman (no use, long-term or short-term methods) based on her demographic and socio-economic characteristics. The sensitive feature is the religion (Islam versus Non-Islam).

DATA / METHOD	CRIME, K = 7		LAW, K = 4		WINE, K = 5		CMC, K = 3	
	Accuracy	Unfairness	Accuracy	Unfairness	Accuracy	Unfairness	Accuracy	Unfairness
reglog + unfair	0.34 ± 0.02	1.12 ± 0.07	0.43 ± 0.01	0.89 ± 0.05	0.54 ± 0.01	0.47 ± 0.05	0.52 ± 0.02	0.78 ± 0.16
reglog + score-fair (baseline)	0.33 ± 0.01	0.78 ± 0.09	0.42 ± 0.01	0.09 ± 0.02	0.54 ± 0.01	0.08 ± 0.03	0.51 ± 0.02	0.25 ± 0.1
reglog + argmax-fair	0.28 ± 0.01	0.26 ± 0.07	0.42 ± 0.01	0.05 ± 0.02	0.54 ± 0.02	0.04 ± 0.01	0.52 ± 0.02	0.19 ± 0.1
linearSVC + unfair	0.36 ± 0.02	1.12 ± 0.07	0.43 ± 0.01	0.97 ± 0.07	0.53 ± 0.01	0.27 ± 0.05	0.51 ± 0.02	0.63 ± 0.22
linearSVC + score-fair (baseline)	0.31 ± 0.02	0.88 ± 0.05	0.42 ± 0.01	0.1 ± 0.03	0.53 ± 0.01	0.1 ± 0.07	0.53 ± 0.02	0.26 ± 0.16
linearSVC + argmax-fair	0.29 ± 0.02	0.25 ± 0.08	0.42 ± 0.01	0.04 ± 0.02	0.53 ± 0.01	0.06 ± 0.04	0.52 ± 0.02	0.2 ± 0.12
GaussSVC + unfair	0.36 ± 0.02	1.4 ± 0.13	0.43 ± 0.01	1.04 ± 0.04	0.53 ± 0.01	0.28 ± 0.06	0.51 ± 0.02	1.0 ± 0.17
GaussSVC + score-fair (baseline)	0.35 ± 0.02	1.02 ± 0.07	0.42 ± 0.01	0.16 ± 0.04	0.55 ± 0.01	0.12 ± 0.04	0.51 ± 0.02	0.16 ± 0.09
GaussSVC + argmax-fair	0.3 ± 0.02	0.22 ± 0.05	0.42 ± 0.01	0.10 ± 0.03	0.55 ± 0.01	0.06 ± 0.03	0.5 ± 0.03	0.2 ± 0.08
RF + unfair	0.37 ± 0.02	1.02 ± 0.04	0.40 ± 0.01	0.65 ± 0.04	0.66 ± 0.01	0.31 ± 0.05	0.55 ± 0.02	0.35 ± 0.18
RF + score-fair (baseline)	0.34 ± 0.02	0.67 ± 0.06	0.39 ± 0.01	0.11 ± 0.05	0.66 ± 0.01	0.09 ± 0.03	0.52 ± 0.03	0.21 ± 0.08
RF + argmax-fair	0.3 ± 0.02	0.33 ± 0.11	0.39 ± 0.01	0.07 ± 0.02	0.66 ± 0.01	0.08 ± 0.02	0.55 ± 0.02	0.22 ± 0.13

Table 5.1: Performance (accuracy & unfairness) of the methods for all datasets and classifiers. We report the means and standard deviations over the 30 repetitions. Colored values highlight fairness.

Results. Results in the multi-class setting are presented in Table 5.1 and highlight the effectiveness of our method. As an example, for the LAW dataset and the GaussSVC with *argmax-fair*, the unfairness is divided by almost 25 (0.97 to 0.04). Furthermore, the *argmax-fair* procedure outperforms the *unfair* and the *score-fair* algorithms for the datasets CRIME, LAW and WINE in terms of unfairness: However, we observe a small decrease of the models accuracy (relatively small compared to the gain in fairness). Note that for the dataset CMC, *score-fair* and *argmax-fair* achieve similar performance.

5.6 Conclusion

Our approach presented in 4 and the alternative method both achieves distribution-free fairness and can be applied on top of any probabilistic base estimator. We illustrate the proficiency of our procedure on various synthetic and real datasets, notably in comparison to the *score-fair* approach suggested in [YX20]. The efficiency of our algorithm in terms of fairness is particularly salient for datasets with large historical bias.

However, our numerical study also outlines the downside of fairness proficiency in terms of classification accuracy. One should hereby be very cautious when using classifiers with strong fairness guarantee, as it possibly degrades the classification quality.

Bibliography

[AAT20] H. Anahideh, A. Asudeh, and S. Thirumuruganathan.
Fair active learning.
arXiv preprint arXiv:2001.01796, 2020.

[AB19] Mohammed Abdelwahab and Carlos Busso.
Active learning for speech emotion recognition using deep neural network.
In *2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII)*, pages 1–7. IEEE, 2019.

[ABD+18a] A. Agarwal, A. Beygelzimer, M. Dudík, J. Langford, and H. Wallach.
A reductions approach to fair classification.
In *Proceedings of the 35th International Conference on Machine Learning*, 2018.

[ABD+18b] Alekh Agarwal, Alina Beygelzimer, Miroslav Dudík, John Langford, and Hanna Wallach.
A reductions approach to fair classification.
In *International Conference on Machine Learning*, pages 60–69. PMLR, 2018.

[ADS+22] Shahriar Akter, Yogesh K Dwivedi, Shahriar Sajib, Kumar Biswas, Ruwan J Bandara, and Katina Michael.
Algorithmic bias in machine learning-based marketing models.
Journal of Business Research, 144:201–216, 2022.

[ADW19] A. Agarwal, M. Dudik, and Z. S. Wu.
Fair regression: Quantitative definitions and reduction-based algorithms.
In *International Conference on Machine Learning*, 2019.

[ALMK16] Julia Angwin, Jeff Larson, Surya Mattu, and Lauren Kirchner.
Machine bias.
ProPublica, May, 23(2016):139–159, 2016.

[AM98] N. Abe and H. Mamitsuka.
Query learning strategies using boosting and bagging.
pages 1–9, 01 1998.

[ARN+21] Md Tofael Ahmed, Maqsudur Rahman, Shafayet Nur, Azm Islam, and Dipankar Das.
Deployment of machine learning and deep learning algorithms in detecting cyberbullying in bangla and romanized bangla text: A comparative study.
In *2021 International Conference on Advances in Electrical, Computing, Communication and Sustainable Technologies (ICAECT)*, pages 1–10. IEEE, 2021.

[AWH18] Bang An, Wenjun Wu, and Huimin Han.
Deep active learning for text classification.
In *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, pages 1–6, 2018.

[BAT+21] Benjamin Gutierrez Becker, Filippo Arcadu, Andreas Thalhammer, Citlalli Gamez Serna, Owen Feehan, Faye Drawnel, Young S Oh, and Marco Prunotto.
Training and deploying a deep learning model for endoscopic severity grading in ulcerative colitis using multicenter clinical trial data.
Therapeutic advances in gastrointestinal endoscopy, 14, 2021.

[BB20] Jeremy R Glissen Brown and Tyler M Berzin.
Deploying artificial intelligence to find the needle in the haystack: deep learning for video capsule endoscopy.
Gastrointestinal Endoscopy, 92(1):152–153, 2020.

[BBL09] M.F. Balcan, A. Beygelzimer, and J. Langford.
Agnostic active learning.
Journal of Computer and System Sciences, 75(1):78–89, 2009.

[BC01] Avrim Blum and Shuchi Chawla.
Learning from labeled and unlabeled data using graph mincuts.
2001.

[BC22] Laurence Barry and Arthur Charpentier.
L'équité de l'apprentissage machine en assurance.
2022.

[BCAR+21] F. Branchaud-Charron, P. Atighehchian, P. Rodríguez, G. Abuhamad, and A. Lacoste.
Can active learning preemptively mitigate fairness issues?
arXiv preprint arXiv:2104.06879, 2021.

[BD98] Kristin Bennett and Ayhan Demiriz.
Semi-supervised support vector machines.
Advances in Neural Information processing systems, 11, 1998.

[Bel58] Richard Bellman.
Dynamic programming and stochastic control processes.
Information and control, 1(3):228–239, 1958.

[Bel66] Richard Bellman.
Dynamic programming.
Science, 153(3731):34–37, 1966.

[BGNK18] W.H. Beluch, T. Genewein, A. Nürnberger, and J.M. Köhler.
The power of ensembles for active learning in image classification.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9368–9377, 2018.

[BHN17] Solon Barocas, Moritz Hardt, and Arvind Narayanan.
Fairness in machine learning.
Nips tutorial, 1:2, 2017.

[BHN18] S. Barocas, M. Hardt, and A. Narayanan.
Fairness and Machine Learning.
fairmlbook.org, 2018.

[BM98] Avrim Blum and Tom Mitchell.
Combining labeled and unlabeled data with co-training.
In *Proceedings of the eleventh annual conference on Computational learning theory*, pages 92–100, 1998.

[BNS06] Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani.
Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.
Journal of machine learning research, 7(11), 2006.

[Bri03] Klaus Brinker.
Incorporating diversity in active learning with support vector machines.
In *Proceedings of the 20th international conference on machine learning (ICML-03)*, pages 59–66, 2003.

[BS14] S. Barocas and A. Selbst.
Big Data's Disparate Impact.
SSRN eLibrary, 2014.

[BS16] Solon Barocas and Andrew D Selbst.
Big data's disparate impact.
Calif. L. Rev., 104:671, 2016.

[BST17] Philip Bachman, Alessandro Sordoni, and Adam Trischler.
Learning algorithms for active learning.
In *international conference on machine learning*, pages 301–310. PMLR, 2017.

[CBP14] Shayok Chakraborty, Vineeth Balasubramanian, and Sethuraman Panchanathan.
Adaptive batch mode active learning.
IEEE transactions on neural networks and learning systems, 26(8):1747–1760, 2014.

[CBS+15] Shayok Chakraborty, Vineeth Balasubramanian, Qian Sun, Sethuraman Panchanathan, and Jieping Ye.
Active batch selection via convex relaxations with guaranteed solution bounds.
IEEE transactions on pattern analysis and machine intelligence, 37(10):1945–1958, 2015.

[CCA+09] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis.
Modeling wine preferences by data mining from physicochemical properties.
Decision Support Systems, 47(4):547–553, 2009.

[CDH+19] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil.
Leveraging labeled and unlabeled data for consistent fair binary classification.
In *Advances in Neural Information Processing Systems*, 2019.

[CDH+20a] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil.
Fair regression via plug-in estimator and recalibration with statistical guarantees.
https://hal.archives-ouvertes.fr/hal-02501190, 2020.

[CDH+20b] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil.
Fair regression via plug-in estimator and recalibrationwith statistical guarantees.
In *Advances in Neural Information Processing Systems*, 2020.

[CDH+20c] E. Chzhen, C. Denis, M. Hebiri, L. Oneto, and M. Pontil.
Fair regression with wasserstein barycenters.
In *Advances in Neural Information Processing Systems*, 2020.

[CG16] T. Chen and C. Guestrin.
Xgboost: A scalable tree boosting system.
In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 785–794, New York, NY, USA, 2016. Association for Computing Machinery.

[Cho17] Alexandra Chouldechova.
Fair prediction with disparate impact: A study of bias in recidivism prediction instruments.
Big data, 5(2):153–163, 2017.

[CJS+20] S. Chiappa, R. Jiang, T. Stepleton, A. Pacchiano, H. Jiang, and J. Aslanides.
A general approach to fairness with optimal transport.
In *AAAI*, 2020.

[CK15] Michał Choraś and Rafał Kozik.
Machine learning techniques applied to detect cyber attacks on web applications.
Logic Journal of the IGPL, 23(1):45–56, 2015.

[CKK+13] T. Calders, A. Karim, F. Kamiran, W. Ali, and X. Zhang.
Controlling attribute effect in linear regression.
In *IEEE International Conference on Data Mining*, 2013.

[CKP09] T. Calders, F. Kamiran, and M. Pechenizkiy.
Building classifiers with independency constraints.
In *IEEE international conference on Data mining*, 2009.

[CKS+18] Elisa Celis, Vijay Keswani, Damian Straszak, Amit Deshpande, Tarun Kathuria, and Nisheeth Vishnoi.
Fair and diverse dpp-based data summarization.
In *International Conference on Machine Learning*, pages 716–725. PMLR, 2018.

[Col07] B. Collins.
Tackling unconscious bias in hiring practices: The plight of the rooney rule.
2007.

[CPC19] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso.
Machine learning interpretability: A survey on methods and metrics.
Electronics, 8(8):832, 2019.

[CS08] Paulo Cortez and Alice Silva.
Using data mining to predict secondary school student performance.
EUROSIS, 01 2008.

[CSZ09] Olivier Chapelle, Bernhard Scholkopf, and Alexander Zien.
Semi-supervised learning (chapelle, o. et al., eds.; 2006)[book reviews].
IEEE Transactions on Neural Networks, 20(3):542–542, 2009.

[CT17] Bo Cowgill and Catherine Tucker.
Algorithmic bias: A counterfactual perspective.
NSF Trustworthy Algorithms, 2017.

[DE95] I. Dagan and S.P. Engelson.
Committee-based sampling for training probabilistic classifiers.

In *In Proceedings of the Twelfth International Conference on Machine Learning*, pages 150–157. Morgan Kaufmann, 1995.

[DEHH21] C. Denis, R. Elie, M. Hebiri, and F. Hu.
Fairness guarantee in multi-class classification, 2021.

[DHP+12] C. Dwork, M. Hardt, T. Pitassi, O. Reingold, and R. Zemel.
Fairness through awareness.
In *Proceedings of the 3rd innovations in theoretical computer science conference*, pages 214–226, 2012.

[DIKL18] C. Dwork, N. Immorlica, A. T. Kalai, and M. D. M. Leiserson.
Decoupled classifiers for group-fair and efficient machine learning.
In *Conference on Fairness, Accountability and Transparency*, 2018.

[DMB16] William Dieterich, Christina Mendoza, and Tim Brennan.
Compas risk scales: Demonstrating accuracy equity and predictive parity.
Northpointe Inc, 7(4), 2016.

[DMDR21] Ankita Dhar, Himadri Mukherjee, Niladri Sekhar Dash, and Kaushik Roy.
Text categorization: past and present.
Artificial Intelligence Review, 54(4):3007–3054, 2021.

[DMF18] Sahil Dhankhad, Emad Mohammed, and Behrouz Far.
Supervised machine learning algorithms for credit card fraudulent transaction detection: a comparative study.
In *2018 IEEE international conference on information reuse and integration (IRI)*, pages 122–125. IEEE, 2018.

[DOBD+18] M. Donini, L. Oneto, S. Ben-David, J. S. Shawe-Taylor, and M. Pontil.
Empirical risk minimization under fairness constraints.
In *Neural Information Processing Systems*, 2018.

[EADvdH17] M Ehsan Abbasnejad, Anthony Dick, and Anton van den Hengel.
Infinite variational autoencoder for semi-supervised learning.
In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5888–5897, 2017.

[EHBG07] S. Ertekin, J. Huang, L. Bottou, and L. Giles.
Learning on the border: Active learning in imbalanced data classification.
In *Proceedings of the Sixteenth ACM Conference on Conference on Information and Knowledge Management*, CIKM '07, page 127–136, New York, NY, USA, 2007. Association for Computing Machinery.

[EHHJ21] Romuald Elie, Caroline Hillairet, François Hu, and Marc Juillard.
An overview of active learning methods for insurance with fairness appreciation.
arXiv preprint arXiv:2112.09466, 2021.

[ESAMS21] Radwa ElShawi, Youssef Sherif, Mouaz Al-Mallah, and Sherif Sakr.
Interpretability in healthcare: A comparative study of local machine learning interpretability techniques.
Computational Intelligence, 37(4):1633–1650, 2021.

[FDM14] Edward W Frees, Richard A Derrig, and Glenn Meyers.
Predictive modeling applications in actuarial science, volume 1.
Cambridge University Press, 2014.

[FFM+15] M. Feldman, S. Friedler, J. Moeller, C. Scheidegger, and S. Venkatasubramanian.
Certifying and removing disparate impact.
In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 259–268. ACM, 2015.

[FLG19] Zhengqing Fu, Goulin Liu, and Lanlan Guo.
Sequential quadratic programming method for nonlinear least squares estimation and its application.
Mathematical Problems in Engineering, 2019.

[FLH07] Ronald N. Forthofer, Eun Sul Lee, and Mike Hernandez.
14 - logistic and proportional hazards regression.
In Ronald N. Forthofer, Eun Sul Lee, and Mike Hernandez, editors, *Biostatistics (Second Edition)*, pages 387–419. Academic Press, San Diego, second edition edition, 2007.

[FS97] Y. Freund and R.E. Schapire.
A decision-theoretic generalization of on-line learning and an application to boosting.
Journal of Computer and System Sciences, 55(1):119–139, 1997.

[FST97] Y. Freund, E. Seung, H.S.and Shamir, and N. Tishby.
Selective sampling using the query by committee algorithm.
Machine Learning, 28:133–168, 1997.

[FZL13] Y. Fu, X. Zhu, and B. Li.
A survey on instance selection for active learning.
Knowledge and information systems, 35(2):249–283, 2013.

[FZTN+12] Rosa L Figueroa, Qing Zeng-Treitler, Long H Ngo, Sergey Goryachev, and Eduardo P Wiechmann.
Active learning for clinical text classification: is it better than random sampling?

Journal of the American Medical Informatics Association, 19(5):809–816, 2012.

[GBR+12] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola.
A kernel two-sample test.
The Journal of Machine Learning Research, 13(1):723–773, 2012.

[GBY+18] Leilani H Gilpin, David Bau, Ben Z Yuan, Ayesha Bajwa, Michael Specter, and Lalana Kagal.
Explaining explanations: An overview of interpretability of machine learning.
In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pages 80–89. IEEE, 2018.

[GCGF16] Gabriel Goh, Andrew Cotter, Maya Gupta, and Michael P Friedlander.
Satisfying real-world goals with dataset constraints.
Advances in Neural Information Processing Systems, 29, 2016.

[GDBFL19] P. Gordaliza, E. Del Barrio, G. Fabrice, and J. M. Loubes.
Obtaining fairness using optimal transport theory.
In *International Conference on Machine Learning*, 2019.

[GG15a] Y. Gal and Z. Ghahramani.
Bayesian convolutional neural networks with bernoulli approximate variational inference.
arXiv preprint arXiv:1506.02158, 2015.

[GG15b] Y. Gal and Z. Ghahramani.
Dropout as a bayesian approximation: Insights and applications.
In *Deep Learning Workshop, ICML*, volume 1, page 2, 2015.

[GG16] Y. Gal and Z. Ghahramani.
Dropout as a bayesian approximation: Representing model uncertainty in deep learning.
In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 1050–1059. JMLR.org, 2016.

[GIG17] Yarin Gal, Riashat Islam, and Zoubin Ghahramani.
Deep bayesian active learning with image data.
In *International Conference on Machine Learning*, pages 1183–1192. PMLR, 2017.

[GKBG18] Mohamed Goudjil, Mouloud Koudil, Mouldi Bedda, and Noureddine Ghoggali.
A novel active learning method using svm for text classification.
International Journal of Automation and Computing, 15(3):290–298, 2018.

[GLR20] T. Gouic, J.M. Loubes, and P. Rigollet.
Projection to fairness in statistical learning.
arXiv preprint arXiv:2005.11720, 2020.

[Gol21] James Goljan.
Developing a cost of delay (cod) framework for the dod & analyzing the current state of air force agile cost estimation.
2021.

[GR08] Shantanu Godbole and Shourya Roy.
Text to intelligence: Building and deploying a text mining solution in the services industry for customer satisfaction analysis.
In *2008 IEEE International Conference on Services Computing*, volume 2, pages 441–448. IEEE, 2008.

[GS07] Yuhong Guo and Dale Schuurmans.
Discriminative batch mode active learning.
Advances in neural information processing systems, 20, 2007.

[GYF18] Naman Goel, Mohammad Yaghini, and Boi Faltings.
Non-discriminatory machine learning through convex fairness criteria.
In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[GYY+20] R Stuart Geiger, Kevin Yu, Yanlai Yang, Mindy Dai, Jie Qiu, Rebekah Tang, and Jenny Huang.
Garbage in, garbage out? do machine learning application papers in social computing report where human-labeled training data comes from?
In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, pages 325–336, 2020.

[Han09] Steve Hanneke.
Theoretical foundations of active learning.
Carnegie Mellon University, 2009.

[HDFMB11] S. Hajian, J. Domingo-Ferrer, and A. Martínez-Ballesté.
Discrimination prevention in data mining for intrusion and crime detection.
In *2011 IEEE Symposium on Computational Intelligence in Cyber Security (CICS)*, pages 47–54, 2011.

[HH00] Harry Holzer and David Holzer.
Assessing affirmative action.
Journal of Economic Literature, 38(3):483–568, 2000.

[HJZ14] Sheng-Jun Huang, Rong Jin, and Zhi-Hua Zhou.
Active learning by querying informative and representative examples.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(10):1936–1949, 2014.

[HJZL06] Steven CH Hoi, Rong Jin, Jianke Zhu, and Michael R Lyu.
Batch mode active learning and its application to medical image classification.
In *Proceedings of the 23rd international conference on Machine learning*, pages 417–424, 2006.

[HKS17] Robin Hirt, Niklas J Koehl, and Gerhard Satzger.
An end-to-end process model for supervised machine learning classification: from problem to deployment in information systems.
In *Designing the Digital Transformation: DESRIST 2017 Research in Progress Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology. Karlsruhe, Germany. 30 May-1 Jun.*, pages 55–63. Karlsruher Institut für Technologie (KIT), 2017.

[HM19] Ben Hutchinson and Margaret Mitchell.
50 years of test (un) fairness: Lessons for machine learning.
In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, pages 49–58, 2019.

[How60] Ronald A Howard.
Dynamic programming and markov processes.
1960.

[HPS16] M. Hardt, E. Price, and N. Srebro.
Equality of opportunity in supervised learning.
In *Neural Information Processing Systems*, 2016.

[HR18] Hielke Hijmans and Charles D Raab.
Ethical dimensions of the gdpr.
Commentary on the General Data Protection Regulation, Cheltenham: Edward Elgar (2018, Forthcoming), 2018.

[HSLZ21] Steven CH Hoi, Doyen Sahoo, Jing Lu, and Peilin Zhao.
Online learning: A comprehensive survey.
Neurocomputing, 459:249–289, 2021.

[HV19] Lingxiao Huang and Nisheeth Vishnoi.
Stable and fair classification.
In *International Conference on Machine Learning*, pages 2879–2890. PMLR, 2019.

[JM15] Michael I Jordan and Tom M Mitchell.
Machine learning: Trends, perspectives, and prospects.
Science, 349(6245):255–260, 2015.

[JN20] Heinrich Jiang and Ofir Nachum.
Identifying and correcting label bias in machine learning.
In *International Conference on Artificial Intelligence and Statistics*, pages 702–712. PMLR, 2020.

[JPS+19] R. Jiang, A. Pacchiano, T. Stepleton, H. Jiang, and S. Chiappa.
Wasserstein fair classification.
arXiv preprint arXiv:1907.12059, 2019.

[KAAS12] Toshihiro Kamishima, Shotaro Akaho, Hideki Asoh, and Jun Sakuma.
Fairness-aware classifier with prejudice remover regularizer.
In *Joint European conference on machine learning and knowledge discovery in databases*, pages 35–50. Springer, 2012.

[Kar81] N El Karoui.
Les aspects probabilistes du contrôle stochastique.
In *École d'été de Probabilités de Saint-Flour IX-1979*, pages 73–238. Springer, 1981.

[KC12] Faisal Kamiran and Toon Calders.
Data preprocessing techniques for classification without discrimination.
Knowledge and information systems, 33(1):1–33, 2012.

[KEUR+18] Anuj Karpatne, Imme Ebert-Uphoff, Sai Ravela, Hassan Ali Babaie, and Vipin Kumar.
Machine learning for the geosciences: Challenges and opportunities.
IEEE Transactions on Knowledge and Data Engineering, 31(8):1544–1554, 2018.

[KGK15] Ilker Kose, Mehmet Gokturk, and Kemal Kilic.
An interactive machine-learning-based electronic fraud and abuse detection system in healthcare insurance.
Applied Soft Computing, 36:283–299, 2015.

[KGM10] Mohit Kumar, Rayid Ghani, and Zhu-Song Mei.
Data mining to predict and prevent errors in health insurance claims processing.
In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 65–74, 2010.

[KGZ19] Michael P Kim, Amirata Ghorbani, and James Zou.
Multiaccuracy: Black-box post-processing for fairness in classification.

In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 247–254, 2019.

[KHE+16] Yoonsang Kim, Jidong Huang, Sherry Emery, et al.
Garbage in, garbage out: data collection, quality assessment and reporting standards for social media data use in health research, infodemiology and digital disease detection.
Journal of medical Internet research, 18(2):e4738, 2016.

[Kir04] Donald E Kirk.
Optimal control theory: an introduction.
Courier Corporation, 2004.

[KKZ12] Faisal Kamiran, Asim Karim, and Xiangliang Zhang.
Decision theory for discrimination-aware classification.
In *2012 IEEE 12th International Conference on Data Mining*, pages 924–929. IEEE, 2012.

[KZC13] F. Kamiran, I. Zliobaite, and T. Calders.
Quantifying explainable discrimination and removing illegal discrimination in automated decision making.
Knowl. Inf. Syst., 35(3):613–644, 2013.

[LBH18] Ming Liu, Wray Buntine, and Gholamreza Haffari.
Learning how to actively learn: A deep imitation learning approach.
In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1874–1883, 2018.

[LGW18] Ismini Lourentzou, Daniel Gruhl, and Steve Welch.
Exploring the efficiency of batch active learning for human-in-the-loop relation extraction.
In *Companion Proceedings of the The Web Conference 2018*, pages 1131–1138, 2018.

[LJ16] K. Lum and J. Johndrow.
A statistical framework for fair predictive algorithms.
arXiv preprint arXiv:1610.08077, 2016.

[LM14] Q. Le and T. Mikolov.
Distributed representations of sentences and documents.
In Eric P. Xing and Tony Jebara, editors, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pages 1188–1196, Bejing, China, 22–24 Jun 2014. PMLR.

[LPB17] B. Lakshminarayanan, A. Pritzel, and C. Blundell.
Simple and scalable predictive uncertainty estimation using deep ensembles.

In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA, 2017. Curran Associates Inc.

[LRT11] Binh Thanh Luong, Salvatore Ruggieri, and Franco Turini.
k-nn as an implementation of situation testing for discrimination discovery and prevention.
In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 502–510, 2011.

[LWCX21] Jian Luo, Jianzong Wang, Ning Cheng, and Jing Xiao.
Loss prediction: End-to-end active learning approach for speech recognition.
In *2021 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2021.

[Mai21] A. Maillart.
Toward an explainable machine learning model for claim frequency: a use case in car insurance pricing with telematics data.
European Actuarial Journal, pages 1–39, 2021.

[MCB20] C. Molnar, G. Casalicchio, and B. Bischl.
Interpretable machine learning – a brief history, state-of-the-art and challenges, 2020.

[MDP+11] Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts.
Learning word vectors for sentiment analysis.
In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[MFA+18] Fausto Milletari, Johann Frei, Moustafa Aboulatta, Gerome Vivar, and Seyed-Ahmad Ahmadi.
Cloud deployment of high-resolution medical image analysis with tomaat.
IEEE journal of biomedical and health informatics, 23(3):969–977, 2018.

[MMS+21] Ninareh Mehrabi, Fred Morstatter, Nripsuta Saxena, Kristina Lerman, and Aram Galstyan.
A survey on bias and fairness in machine learning.
ACM Computing Surveys (CSUR), 54(6):1–35, 2021.

[MN98] A.K. McCallumzy and K. Nigamy.
Employing em and pool-based active learning for text classification.
In *Proc. International Conference on Machine Learning (ICML)*, pages 359–367. Citeseer, 1998.

[MPH+17] Alireza Mehrtash, Mehran Pesteie, Jorden Hetherington, Peter A Behringer, Tina Kapur, William M Wells III, Robert Rohling, Andriy Fedorov, and Purang Abolmaesumi.
Deepinfer: Open-source deep learning deployment toolkit for image-guided therapy.
In *Medical Imaging 2017: Image-Guided Procedures, Robotic Interventions, and Modeling*, volume 10135, pages 410–416. SPIE, 2017.

[MSC+13] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean.
Distributed representations of words and phrases and their compositionality.
In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA, 2013. Curran Associates Inc.

[MU96] David J Miller and Hasan Uyar.
A mixture of experts classifier with learning based on both labelled and unlabelled data.
Advances in neural information processing systems, 9, 1996.

[MWW+18] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, and Joel T Dudley.
Deep learning for healthcare: review, opportunities and challenges.
Briefings in bioinformatics, 19(6):1236–1246, 2018.

[Nar18] Harikrishna Narasimhan.
Learning with complex loss functions and constraints.
In *International Conference on Artificial Intelligence and Statistics*, pages 1646–1654. PMLR, 2018.

[Nes83] Y. Nesterov.
A method for solving the convex programming problem with convergence rate O(1/kˆ2).
In *Doklady Akademii Nauk SSSR*, 1983.

[Nes13] Y. Nesterov.
Introductory lectures on convex optimization: A basic course, volume 87.
Springer Science & Business Media, 2013.

[Nie07] Pu-yan Nie.
Sequential penalty quadratic programming filter methods for nonlinear programming.
Nonlinear Analysis: Real World Applications, 8(1):118–129, 2007.

[NMS12] Udhyakumar Nallasamy, Florian Metze, and Tanja Schultz.
Active learning for accent adaptation in automatic speech recognition.
In *2012 IEEE Spoken Language Technology Workshop (SLT)*, pages 360–365. IEEE, 2012.

[NMTM00a] Kamal Nigam, Andrew McCallum, Sebastian Thrun, and Tom M. Mitchell.
Text classification from labeled and unlabeled documents using em.

Machine Learning, 39(2/3):103–134, 2000.

[NMTM00b] Kamal Nigam, Andrew Kachites McCallum, Sebastian Thrun, and Tom Mitchell.
Text classification from labeled and unlabeled documents using em.
Machine learning, 39(2):103–134, 2000.

[NVK+15] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic.
Deep learning applications and challenges in big data analytics.
Journal of big data, 2(1):1–21, 2015.

[ODP19] L. Oneto, M. Donini, and M. Pontil.
General fair empirical risk minimization.
arXiv preprint arXiv:1901.10080, 2019.

[OHT20] Yassine Ouali, Céline Hudelot, and Myriam Tami.
An overview of deep semi-supervised learning.
arXiv preprint arXiv:2006.05278, 2020.

[otPMD+16] Executive Office of the President, Cecilia Munoz, Domestic Policy Council Director, Megan (US Chief Technology Officer Smith (Office of Science, Technology Policy)), DJ (Deputy Chief Technology Officer for Data Policy, Chief Data Scientist Patil (Office of Science, and Technology Policy)).
Big data: A report on algorithmic systems, opportunity, and civil rights.
Executive Office of the President, 2016.

[PB12] Swarnajyoti Patra and Lorenzo Bruzzone.
A cluster-assumption based batch mode active learning technique.
Pattern Recognition Letters, 33(9):1042–1048, 2012.

[PDWH18] Kunkun Pang, Mingzhi Dong, Yang Wu, and Timothy Hospedales.
Meta-learning transferable active learning policies by deep reinforcement learning.
arXiv preprint arXiv:1806.04798, 2018.

[Pha09] Huyên Pham.
Continuous-time stochastic control and optimization with financial applications, volume 61.
Springer Science & Business Media, 2009.

[PMM18] Marcos Pacheco, Antoni-Lluís Mesquida, and Antònia Mas.
Being agile while coaching teams using their own data.
In *European Conference on Software Process Improvement*, pages 426–436. Springer, 2018.

[PRW+17] G. Pleiss, M. Raghavan, F. Wu, J. Kleinberg, and K. Weinberger.

On fairness and calibration.
In *Neural Information Processing Systems*, 2017.

[Rei09] Donald G Reinertsen.
The principles of product development flow: second generation lean product development, volume 62.
Celeritas Redondo Beach, 2009.

[RHW19] Yuji Roh, Geon Heo, and Steven Euijong Whang.
A survey on data collection for machine learning: a big data-ai integration perspective.
IEEE Transactions on Knowledge and Data Engineering, 33(4):1328–1347, 2019.

[RLC+21] Dan Jeric Arcega Rustia, Chen-Yi Lu, Jun-Jee Chao, Ya-Fang Wu, Jui-Yung Chung, Ju-Chun Hsu, and Ta-Te Lin.
Online semi-supervised learning applied to an automated insect pest monitoring system.
Biosystems Engineering, 208:28–44, 2021.

[RM01] N. Roy and A. McCallum.
Toward optimal active learning through sampling estimation of error reduction.
In *ICML*, 2001.

[RMC15] Alec Radford, Luke Metz, and Soumith Chintala.
Unsupervised representation learning with deep convolutional generative adversarial networks.
arXiv preprint arXiv:1511.06434, 2015.

[RP11] S Benson Edwin Raj and A Annie Portia.
Analysis on credit card fraud detection methods.
In *2011 International Conference on Computer, Communication and Electrical Technology (ICCCET)*, pages 152–156. IEEE, 2011.

[RS13] T. Reitmaier and B. Sick.
Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4ds.
Information Sciences, 230:106–131, 05 2013.

[Rub99] R. Rubinstein.
The cross-entropy method for combinatorial and continuous optimization.
Methodology and computing in applied probability, 1(2):127–190, 1999.

[RXC+20] P.n Ren, Y. Xiao, X. Chang, P.Y. Huang, Z. Li, X. Chen, and X. Wang.
A survey of deep active learning.
arXiv preprint arXiv:2009.00236, 2020.

[RZS19] Yongjian Ren, Kun Zhang, and Yuliang Shi.
A survival certification model based on active learning over medical insurance data.
In Jie Shao, Man Lung Yiu, Masashi Toyoda, Dongxiang Zhang, Wei Wang, and Bin Cui, editors, *Web and Big Data*, pages 156–170, Cham, 2019. Springer International Publishing.

[SC00] Greg Schohn and David Cohn.
Less is more: Active learning with support vector machines.
In *ICML*, volume 2, page 6. Citeseer, 2000.

[SC08a] B. Settles and M. Craven.
An analysis of active learning strategies for sequence labeling tasks.
In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '08, page 1070–1079, USA, 2008. Association for Computational Linguistics.

[SC08b] Burr Settles and Mark Craven.
An analysis of active learning strategies for sequence labeling tasks.
In *proceedings of the 2008 conference on empirical methods in natural language processing*, pages 1070–1079, 2008.

[SDI20] A. Sharaf and H. Daumé III.
Promoting fairness in learned models by learning to active learn under parity constraints.
In *ICML Workshops*, 2020.

[SDP18] Giorgio Alfredo Spedicato, Christophe Dutang, and Leonardo Petrini.
Machine learning methods to perform pricing optimization. a comparison with standard glms.
Variance, 12(1):69–89, 2018.

[SED19] S. Sinha, S. Ebrahimi, and T. Darrell.
Variational adversarial active learning.
In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5972–5981, 2019.

[Set09] B. Settles.
Active learning literature survey.
2009.

[SGCA18] Eishvak Sengupta, Dhruv Garg, Tanupriya Choudhury, and Archit Aggarwal.
Techniques to elimenate human bias in machine learning.
2018 International Conference on System Modeling & Advancement in Research Trends (SMART), pages 226–230, 2018.

[Sha48] C.E. Shannon.
A mathematical theory of communication.
The Bell system technical journal, 27(3):379–423, 1948.

[SNL+18] J.S. Smith, B. Nebgen, N. Lubbers, O. Isayev, and A.E. Roitberg.
Less is more: Sampling chemical space with active learning.
The Journal of chemical physics, 148(24):241733, 2018.

[SNZ08] Aarti Singh, Robert Nowak, and Jerry Zhu.
Unlabeled data: Now it helps, now it doesn't.
In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems*, volume 21. Curran Associates, Inc., 2008.

[SOS92] H. Seung, M. Opper, and H. Sompolinsky.
Query by committee.
pages 287–294, 01 1992.

[SR10] Kevin Small and Dan Roth.
Margin-based active learning for structured predictions.
International Journal of Machine Learning and Cybernetics, 1(1):3–25, 2010.

[SS+11] S. Shalev-Shwartz et al.
Online learning and online convex optimization.
Foundations and trends in Machine Learning, 4(2):107–194, 2011.

[SS17] Hillary Sanders and Joshua Saxe.
Garbage in, garbage out: how purportedly great ml models can be screwed up by bad data.
Proceedings of Blackhat, 2017, 2017.

[SS18] O. Sener and S. Savarese.
Active learning for convolutional neural networks: A core-set approach.
In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018.

[SSD18] Jonathon Stewart, Peter Sprivulis, and Girish Dwivedi.
Artificial intelligence and machine learning in emergency medicine.
Emergency Medicine Australasia, 30(6):870–874, 2018.

[ST11] S. Sivaraman and M. Trivedi.
Active learning for on-road vehicle detection: a comparative study.
Machine Vision and Applications, 25:599–611, 2011.

[THN+18] Welderufael B Tesfay, Peter Hofmann, Toru Nakamura, Shinsaku Kiyomoto, and Jetzabel Serna.
I read but don't agree: Privacy policy benchmarking using machine learning and the eu gdpr.
In *Companion Proceedings of the The Web Conference 2018*, pages 163–166, 2018.

[TK01] Simon Tong and Daphne Koller.
Support vector machine active learning with applications to text classification.
Journal of machine learning research, 2(Nov):45–66, 2001.

[TRN20] Shiv Kumar Tavker, Harish Guruprasad Ramaswamy, and Harikrishna Narasimhan.
Consistent plug-in classifiers for complex objectives and constraints.
In *Advances in Neural Information Processing Systems*, volume 33, pages 20366–20377, 2020.

[TS10] Lisa Torrey and Jude Shavlik.
Transfer learning.
In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*, pages 242–264. IGI global, 2010.

[TYV16] Dimitrios I. Tselentis, George Yannis, and Eleni I. Vlahogianni.
Innovative insurance schemes: Pay as/how you drive.
Transportation Research Procedia, 14:362–371, 2016.
Transport Research Arena TRA2016.

[VC15] V. Vapnik and A Chervonenkis.
On the uniform convergence of relative frequencies of events to their probabilities.
In *Measures of complexity*, pages 11–30. Springer, 2015.

[VEH20] Jesper E Van Engelen and Holger H Hoos.
A survey on semi-supervised learning.
Machine Learning, 109(2):373–440, 2020.

[WCW+17] Mowei Wang, Yong Cui, Xin Wang, Shihan Xiao, and Junchen Jiang.
Machine learning for networking: Workflow, advances and opportunities.
Ieee Network, 32(2):92–99, 2017.

[WKW16] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang.
A survey of transfer learning.
Journal of Big data, 3(1):1–40, 2016.

[WR98] L. F. Wightman and H. Ramsey.
LSAC national longitudinal bar passage study.

Law School Admission Council, 1998.

[WWZJ19] Defu Wang, Xiaojuan Wang, Yong Zhang, and Lei Jin.
Detection of power grid disturbances and cyber-attacks based on machine learning.
Journal of information security and applications, 46:42–52, 2019.

[WZL+16] Keze Wang, Dongyu Zhang, Ya Li, Ruimao Zhang, and Liang Lin.
Cost-effective active learning for deep image classification.
IEEE Transactions on Circuits and Systems for Video Technology, 27(12):2591–2600, 2016.

[WZS19] Hanmo Wang, Runwu Zhou, and Yi-Dong Shen.
Bounding uncertainty for active batch selection.
In *Proceedings of the AAAI conference on artificial intelligence*, volume 33, pages 5240–5247, 2019.

[XJZ+09] Zenglin Xu, Rong Jin, Jianke Zhu, Irwin King, Michael Lyu, and Zhirong Yang.
Adaptive regularization for transductive support vector machine.
Advances in Neural Information Processing Systems, 22, 2009.

[Yar95] David Yarowsky.
Unsupervised word sense disambiguation rivaling supervised methods.
In *33rd annual meeting of the association for computational linguistics*, pages 189–196, 1995.

[YCK19] Yang Yuan, Soo-Whan Chung, and Hong-Goo Kang.
Gradient-based active learning query strategy for end-to-end speech recognition.
In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2832–2836. IEEE, 2019.

[YMN+15] Yi Yang, Zhigang Ma, Feiping Nie, Xiaojun Chang, and Alexander G Hauptmann.
Multi-class active learning by uncertainty sampling with diversity maximization.
International Journal of Computer Vision, 113(2):113–127, 2015.

[YP11] Ting Yang and Carey E. Priebe.
The effect of model misspecification on semi-supervised classification.
IEEE Transactions on Pattern Analysis and Machine Intelligence, 33(10):2093–2103, 2011.

[YQC+17] Changchang Yin, Buyue Qian, Shilei Cao, Xiaoyu Li, Jishang Wei, Qinghua Zheng, and Ian Davidson.
Deep similarity-based batch mode active learning with exploration-exploitation.
In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 575–584, 2017.

[YX20] Q. Ye and W. Xie.

Unbiased subdata selection for fair classification: A unified framework and scalable algorithms.
arXiv preprint arXiv:2012.12356, 2020.

[ZG09] Xiaojin Zhu and Andrew B Goldberg.
Introduction to semi-supervised learning.
Synthesis lectures on artificial intelligence and machine learning, 3(1):1–130, 2009.

[ZGL03] Xiaojin Zhu, Zoubin Ghahramani, and John D Lafferty.
Semi-supervised learning using gaussian fields and harmonic functions.
In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, pages 912–919, 2003.

[ZH07] J. Zhu and E. Hovy.
Active learning for word sense disambiguation with methods for addressing the class imbalance problem.
In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 783–790, 2007.

[Zha04] T. Zhang.
Statistical behavior and consistency of classification methods based on convex risk minimization.
The Annals of Statistics, 32, 2004.

[Zhu05] Xiaojin Jerry Zhu.
Semi-supervised learning literature survey.
2005.

[ZSAT19] Ameema Zainab, Dabeeruddin Syed, and Dena Al-Thani.
Deployment of deep learning models to mobile devices for spam classification.
In *2019 IEEE First International Conference on Cognitive Machine Intelligence (CogMI)*, pages 112–117. IEEE, 2019.

[ZVGRG17] M. B. Zafar, I. Valera, M. Gomez Rodriguez, and K. P. Gummadi.
Fairness beyond disparate treatment & disparate impact: Learning classification without disparate mistreatment.
In *International Conference on World Wide Web*, 2017.

[ZVGRG19] M. B. Zafar, I. Valera, M. Gomez-Rodriguez, and K. P. Gummadi.
Fairness constraints: A flexible approach for fair classification.
Journal of Machine Learning Research, 20(75):1–42, 2019.

[ZWLD16] Guoqiang Zhong, Li-Na Wang, Xiao Ling, and Junyu Dong.
An overview on data representation learning: From traditional feature learning to recent deep learning.
The Journal of Finance and Data Science, 2(4):265–278, 2016.

[ZWS+13] R. Zemel, Y. Wu, K. Swersky, T. Pitassi, and C. Dwork.
Learning fair representations.
In *International Conference on Machine Learning*, 2013.

[ZWY+20] Ling Zhang, Xiaosong Wang, Dong Yang, Thomas Sanford, Stephanie Harmon, Baris Turkbey, Bradford J Wood, Holger Roth, Andriy Myronenko, Daguang Xu, et al.
Generalizing deep learning for medical image segmentation to unseen domains via deep stacked transformation.
IEEE transactions on medical imaging, 39(7):2531–2540, 2020.

[ZXX18] Pei Zhang, Xueying Xu, and Deyi Xiong.
Active learning for neural machine translation.
In *2018 International Conference on Asian Language Processing (IALP)*, pages 153–158. IEEE, 2018.

List of Tables

List of Figures

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Apprentissage semi-supervisé en assurance : équité et apprentissage actif

Mots clés : Actuariat, Classification, Apprentissage Actif, Équité, Traitement Automatique des Langues

Résumé : Les organismes d'assurance stockent quotidiennement des sources de données textuelles volumineuses (zones de texte libre utilisées par les téléconseillers, courriers électroniques, avis des clients, etc.). Cependant, cette masse de données textuelles comporte des enjeux spécifiques en termes de réglementations comme par exemple le respect des contraintes de protection de la vie privée, imposées en Europe par le récent Règlement général sur la protection des données (RGPD) : ces données textuelles peuvent contenir des informations non-conformes aux normes RGPD, soulevant ainsi des questions éthiques et ne peuvent pas être conservées par l'assureur. Aujourd'hui, ces données textuelles sont étiquetées par des experts (oracles) et ce processus n'est pas adapté à la gestion de grands volumes ni à une gestion de l'information en temps quasi réel. Par conséquent, la mise en place d'un système d'apprentissage précis (en termes de prédiction), peu coûteux (en termes d'étiquetage) et éthique (en termes d'équité) est nécessaire en assurance et cette thèse aborde et résout certains de ces défis. Comme les données non étiquetées sont généralement abondantes dans le secteur de l'assurance, le premier défi consiste à réduire l'effort d'étiquetage grâce à l'apprentissage actif, une boucle de rétroaction entre l'inférence du modèle et un oracle. Un autre défi majeur est la question de l'équité dans les inférences des modèles d'apprentissage. Puisque des inégalités et des discriminations peuvent être trouvées dans les données, les modèles d'apprentissage sont susceptibles de reproduire certaines injustices, ce qui les rend inutilisables en production. Cette thèse explore ces problèmes et propose des solutions, notamment pour les tâches de classification multi-classes. En particulier, nous proposons une méthode d'équité algorithmique qui garantit soit une équité exacte au détriment de la précision du modèle, soit un compromis entre équité et précision appelé epsilon-fairness. En outre, nous proposons une méthode d'apprentissage actif équitable qui requête les instances informatives tout en rendant le modèle équitable. Enfin, nous proposons des stratégies d'étiquetage optimales qui réduisent considérablement l'effort humain tout en gardant de bonnes performances du modèle. Toutes les méthodologies proposées ont l'avantage d'être agnostiques au modèle d'apprentissage. Les résultats sont étudiés et appliqués sur des jeux de données réels et synthétiques.

Title : Semi-supervised learning in insurance : fairness and active learning

Keywords : Actuarial Science, Classification, Active Learning, Fairness, Natural Language Processing

Abstract : Insurance organisations store voluminous textual data sources on a daily basis (free text fields used by telephonists, emails, customer reviews, ...). However, this mass of textual data involves specific issues in terms of regulations, such as compliance with the privacy constraints imposed in Europe by the recent General Data Protection Regulation (GDPR) : this textual data may contain information that is not compliant with the GDPR standards, thus raising ethical issues and cannot be retained by the insurer. Today, this textual data is tagged by experts (oracles) and this process is not suitable for managing large volumes and near real-time information. Therefore, the implementation of an accurate (in terms of prediction), low-cost (in terms of labelling) and ethical (in terms of fairness) learning system is needed in insurance and this thesis addresses and solves some of these challenges. As unlabelled data in insurance is usually abundant, the first challenge is to reduce the labelling effort through active learning, a feedback loop between model inference and an oracle. Another major challenge is the issue of fairness in machine learning model inferences. Since inequalities and discriminations can be found in the data, learning models are likely to reproduce some unfairness, making them unusable in production. This thesis explores these problems and proposes solutions, especially for multi-class classification tasks. In particular, we propose an algorithmic fairness method that guarantees either exact fairness at the expense of model accuracy, or a compromise between fairness and accuracy called epsilon-fairness. In addition, we propose a fair active learning method that requests informative instances while making the model fair. Finally, we propose optimal labelling strategies that significantly reduce human effort while maintaining good model performance. All the proposed methodologies have the advantage of being agnostic to the learning model. The results are studied and applied on real and synthetic datasets.

Institut Polytechnique de Paris
91120 Palaiseau, France