AVANTIX	Etat de l'art sur l'Active Learning et le Deep Active Learning	DP073778NTE001 Version 01 Date : 11/04/2022 1/33 Confidentiel

Code Affaire : **1-10-R&D-2007**

N° Marché ou N° Cde : **-**

Classification : **DR-SF**

Diffusion interne : -

Diffusion externe : -

Rédigé par	Vérifié par	Approuvé par	Approuvé par
R. MAZOUZ	**C. BELAFDIL**		
Visa	Visa	Visa	Visa

En l'absence de tout retour formel dans un délai maximum de 3 semaines, ce document est considéré comme approuvé par défaut par le client.

Nombre de pages (hors annexes) : **23**

Etat de l'art sur l'Active Learning et le Deep Active Learning

EVOLUTIONS

IND.	DATE	SECR.	REDACTEURS	OBJET
00	**11/04/22**	**xxx**	**R. MAZOUZ**	**Version Initiale**

SOMMAIRE

Liste de figures

Liste de tableaux

Liste d'équations

OBJET

Ce document présente un état de l'art pour l'identification de signaux sur la base de méthodes d'Active Learning. Cet état de l'art s'inscrit dans le contexte du projet DESPINA qui a pour objet l'identification d'émetteurs (radars) à partir des produits d'interception CERES (PIC) qui sont construits à partir des scènes électromagnétiques observées. Néanmoins cet état de l'art se veut réutilisable pour d'autres projets qui nécessiteraient l'utilisation de méthodes d'Active Learning.

2. DOCUMENTS APPLICABLES

Rep.	Intitulé	Référence	Indice	Date
[DA0]	Cahier des Clauses Techniques Particulières DESPINA (CCTP)	16-054199/DO/UM ESIO/SRE/DRSF	0.1	09/01/2020

3. DOCUMENTS DE REFERENCE

Rep.	Intitulé	Référence	Indice	Date
[DR1]	Manuel Qualité AVANTIX	96QP007	25	
[DR2]	Certificats ISO9001 et EN9100 AVANTIX	/	/	
[DR3]	Spécification Technique de Besoin (STB) DESPINA	DP073065STB001	03	05/11/2020

4. GLOSSAIRE ET ABREVIATIONS

Terme	*Définition*
AL	Active Learning est une méthode d'apprentissage semi-supervisé où un oracle intervient au cours du processus. A la différence des autres méthodes d'apprentissage en Machine et Deep Learning, en Active Learning c'est l'algorithme d'apprentissage qui demande des informations pour l'obtention de certaines données précises.
BNN	Bayesian Neural Network
Budget	Nombre maximum de requêtes autorisées
CNN	Convolutional Neural Network
Decision Tree	Decision Tree (arbre de décision) est un algorithme permettant de classer en construisant des seuils optimaux sur les caractéristiques d'entrées. Algorithme de base utilisé par le Random Forest.
DeepAL	Deep Active Learning est une méthode d'active learning utilisant des modèles de Deep Learning.
IA	Intelligence Artificielle
k-NN	K Nearest Neighbors (k plus proches voisins) est un algorithme de type « instance based » ; lors de l'étape d'inférence l'entrée est étiquetée selon les étiquettes de ses k plus proches voisins (habituellement au sens de la distance euclidienne) par vote majoritaire.
LSTM	Long Short-Term Memory est un réseau de neurones de type récurrent (voir RNN) qui a deux sorties, l'une qui transmet l'information court-terme et l'autre long-terme. Réseau qui ne corrige qu'en partie l'absence de mémoire à long-terme.
MLP	Multi-Layer Perceptron est un type de réseau composé uniquement de perceptrons agencés en couches. Il n'y a pas de définition universelle mais la communauté s'accorde à dire qu'un réseau est profond (deep) quand il possède au moins deux couches cachées.
Oracle	Humain ou machine labelisant la donnée
Outlier	Un outlier pourrait se traduire par : donnée aberrante. C'est une valeur ou une observation qui est « distante » des autres observations effectuées sur le même phénomène.
Overfitting	Le terme d'overfitting est employé pour décrire le surapprentissage d'un modèle statistique lors d'un apprentissage automatique.
Perceptron	La plus simple des architectures d'un réseau de neurones artificiel, celui-ci ne possède qu'une seule couche et qu'une seule sortie. Il connecte la sortie aux entrées en appliquant une combinaison linéaire des entrées (à partir des poids w_i), l'ajout d'un biais (scalaire noté b) et une fonction d'activation qui doit être non-linéaire, différentiable et monotone (notée σ). La formule suivante résume le calcul réalisé par un perceptron : $o = \sigma(\sum_{i=1}^{N} x_i . w_i + b)$ où o est la sortie et N le nombre d'entrées.
Random Forest	Random Forest (Forêt aléatoire) est un algorithme permettant de créer plusieurs arbres choisissant aléatoirement les caractéristiques à utiliser pour chacun d'eux. Voir Decision Tree.
Requête	Demande d'étiquetage d'une donnée envoyée par l'Oracle
RNN	Recurrent Neural Network est un type de réseau capable de traiter des données d'entrées séquentielles. La sortie du réseau est concaténée avec la séquence d'entrée suivante. RNN fait référence au réseau éponyme le plus triviale mais également à tous types de réseaux récurrents comme le LSTM et le GRU.

Terme	*Définition*
Stopping Criteria	Condition stoppant le processus d'Active Learning
SVM	Support Vector Machine. Algorithme de classification très utilisé pour ses bonnes performances et sa rapidité d'exécution. Son principe est de rechercher des hyperplans séparateurs entre les classes qui maximisent la marge. Le SVM a pour principal défaut de ne pas converger lorsque le nombre d'entrée est grand. Le SVM a été pensé pour résoudre des problèmes linéaires mais grâce au « kernel trick », l'algorithme peut résoudre les problèmes non linéaires.

5. CONTEXTE

Dans un monde où la donnée est de plus en plus accessible en grande quantité, les algorithmes de Machine Learning et plus récemment ceux de Deep Learning sont devenus une référence dans le monde de la science des données. Néanmoins, obtenir des données correctement labélisées est très souvent couteux et peut devenir une réelle contrainte dans l'entraînement de ces algorithmes. L'Active Learning (parfois nommé « apprentissage actif »), premièrement pensé dans les années 90 qui par la suite s'est vu être peu exploré, est aujourd'hui de plus en plus un sujet d'intérêt pour les data scientist et tend à minimiser la quantité de données labélisées à fournir dans le processus d'entrainement des différents modèles.
Nous réaliserons un état de l'art de l'Active Learning et du Deep Active Learning.

L'Active Learning est un procédé semi-supervisé où un oracle est sollicité pour labéliser des données. L'objectif est de maximiser les performances d'un modèle de Machine Learning en ayant minimisé la quantité de données labélisées fournies lors du processus d'apprentissage.
Les principes fondamentaux de l'Active Learning sont :

- Certaines données sont plus importantes que d'autres lors du processus d'apprentissage
- Labéliser une donnée n'est pas nul en coût

6. SCENARIOS ET STRATEGIES D'ACTIVE LEARNING ET DEEP ACTIVE LEARNING

Les parties essentielles de cette section sont :

6.1. SCENARIOS

Plusieurs processus d'apprentissage peuvent nécessiter l'utilisation de l'Active Learning. Par la suite nous allons définir les trois types de situations possibles en Active Learning.

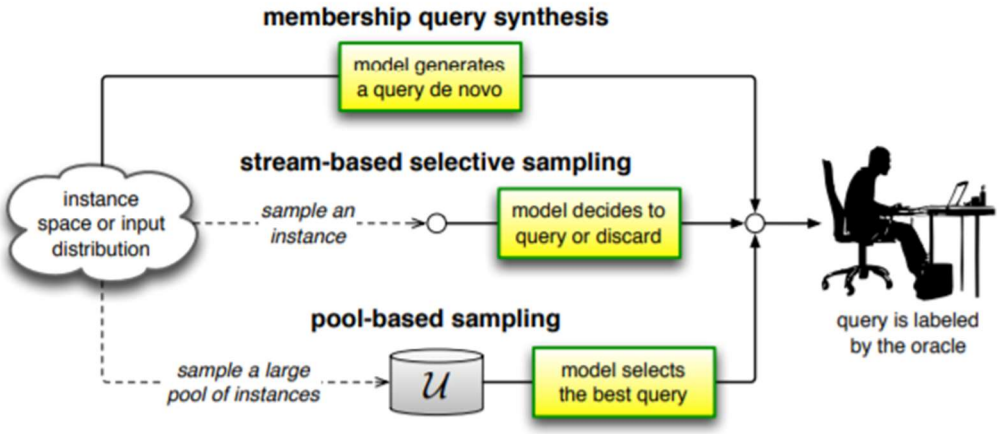


Figure 1 : DIAGRAMME PRESENTANT LES TROIS SCENARIOS PRINCIPAUX DE L'ACTIVE LEARNING QUI SONT LE MEMBERSHIP QUERY SNTHESIS, LE STREAM BASED SELECTIVE SAMPLING, LE POOL BASED SAMPLING. SOURCE : (BURR SETTLES, 2010) [1]

6.1.1. MEMBERSHIP QUERY STRATEGY

Le premier type est le *Membership Query Strategy* (Synthèse de Requête). Dans ce procédé, l'apprenant va créer lui-même les requêtes sur les espaces les plus ambigus de l'espace de donnée. Cette stratégie suppose qu'il est possible de labéliser des données non répertoriées. Les résultats peuvent être confus car il arrive que l'Oracle rencontre des difficultés pour labéliser une donnée crée par l'algorithme, par exemple lors de la labélisation des images lorsque celles-ci sont créées de façon artificielle. A contrario, cette stratégie peut se révéler prometteuse dans des situations où le processus de récolte de la donnée est simple et où l'Oracle sait labéliser n'importe quelle donnée, par exemple lorsqu'il s'agit d'identifier la couleur d'un point (x,y) répertorié sur un plan.

6.1.2. STREAM BASED SELECTIVE SAMPLING

Aussi nommé « apprentissage en ligne », le *Stream Based Selective Sampling* est une méthode où des données non labélisées arrivent successivement et l'algorithme d'Active Learning décide si labéliser la donnée est pertinent ou non. Cette méthode peut être efficace lorsqu'on veut réduire l'espace mémoire nécessaire pour l'apprentissage par exemple dans le cas d'appareils embarqués.

6.1.3. POOL BASED SAMPLING

Le Pool Based Sampling ou Apprentissage hors ligne est la technique la plus couramment utilisée. Elle est pertinente lorsqu'accéder à des données non labélisées est simple. Dans ce procédé, on va fournir l'ensemble des données non labélisées à l'apprenant qui va déterminer celles qui sont les plus pertinentes. L'algorithme d'Active Learning ayant accès à l'ensemble des données et non à un flux de donnée, aura théoriquement plus de facilité à effectuer la meilleure requête. Ainsi, le nombre de requête à réaliser peut diminuer drastiquement.

6.2. QUERY STRATEGIES POUR L'ACTIVE LEARNING

Il existe différentes stratégies permettant de sélectionner les données nécessitant d'être labéliser par l'Oracle. La littérature présente six strategies distinctes d'Active Learning : *Uncertainty Sampling*, *Query By Committee*, *Expected Model Change*, *Expected Error Reduction*, *Variance Reduction* et *Density Weighted Methods*. Dans ce qui suit nous allons tout d'abord énoncer le problème et lister les variables qui nous seront utiles. Puis nous allons détailler le processus de ces différentes stratégies. Enfin nous récapitulerons leurs avantages et leurs inconvénients respectifs.

6.2.1. ENONCE DU PROBLEME

Dans le contexte d'un problème d'Active Learning et dans le scénario de *Pool Based Sampling 6.1.3*, nous sommes initialement en possession d'un grand nombre de données non labélisées, potentiellement d'un ensemble de données initialement labélisées et d'un modèle de Machine Learning permettant la prédiction du label. L'objectif est de sélectionner quelques données afin de les labéliser et permettre à l'algorithme de Machine Learning d'améliorer au mieux sa prédiction.

Les variables utilisées sont :

- $L = \{ x_1, x_2, \dots x_l \}$, $l \geq 0$, l'ensemble des données labélisées.
  $U = \{ x_1^u, x_1^u, \dots x_n^u \}$, $n \geq 1$, l'ensemble des données non labélisées.
- $Y = \{y_1^u, y_2^u, \dots, y_n^u\}$ est l'ensemble des labels
- $\hat{Y} = \{\hat{y}_1^u, \hat{y}_2^u, \dots, \hat{y}_n^u\}$ est l'ensemble des labels prédit par le modèle θ.
- $P_\theta(y|x)$ est la probabilité de labéliser x par y selon le modèle θ.

$$x^* \text{ est la donnée nécessitant la labélisation.}$$

6.2.2. UNCERTAINTY SAMPLING

La stratégie *Uncertainty Sampling* (échantillonnage par incertitude) tente de déterminer les données qui ont le plus de mal à être prédite par le modèle de Machine Learning. Il existe là aussi plusieurs méthodes permettant de repérer cette donnée :

- *Least Confident* (LC) : On va déterminer la requête à effectuer en fonction du score de confiance lors de la prédiction réalisée par le modèle. Plus précisément, on va sélectionner la donnée où la probabilité de prédiction à une certaine classe est la plus faible.

Équation 1 : Least Confident Strategy

$$x^* = argmax_x(1 - P_\theta(\hat{y}|x))$$

Avec $\hat{y}$ le label ayant la plus haute probabilité d'être attribué à la donnée x

- Margin Sampling : Le principe est de sélectionner la donnée dont la différence entre les probabilités de prédiction des deux classes les plus probables est la plus haute. On cherche donc à sélectionner la donnée dont le modèle serait « hésitant » dans sa prédiction.

Équation 2 : Margin Sampling Strategy

$$x^* = argmax_x(P_\theta(\hat{y}_1|x) - P_\theta(\hat{y}_2|x))$$

Avec $\hat{y}_1$ et $\hat{y}_2$ les labesl ayant les plus hautes probabilités d'être attribué à la donnée x

- Entropy Sampling : Cette technique se base sur le calcul de l'entropie introduit par Claude Shannon, lors de chaque prédiction. On va tenter de minimiser cette entropie.

Équation 3 : Calcul de l'entropie

$$x^* = argmax_x(1 - \sum_i P_\theta(\hat{y}_i|x)\, log\big(P_\theta(\hat{y}_i|x)\big))$$

La particularité de cette méthode est qu'elle permet de calculer le « doute » dans la prédiction du modèle sur l'ensemble des prédictions réalisées et non plus uniquement sur les deux plus probables tel que dans la technique du Margin Sampling.

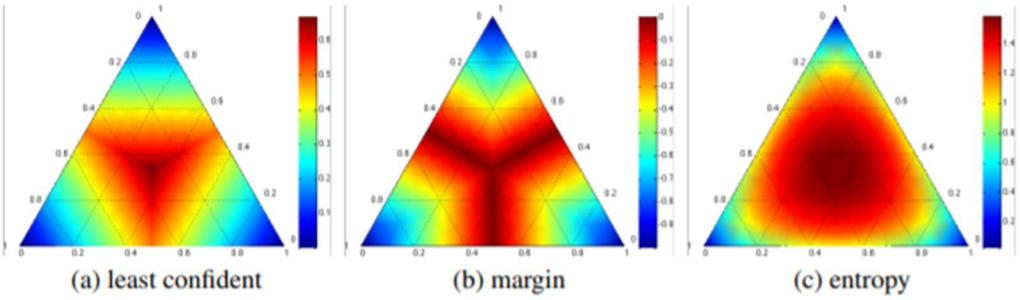


Figure 2 : EXEMPLE PERMETTANT DE VISUALISER L'IMPACT DES DIFFERENTES TECHNIQUES DE « QUERY STRATEGY » SUR LA SELECTION DE LA DONNEE PERTINENTE DANS UN MODELE DE SELECTION ENTRE TROIS CLASSES. PLUS LA DONNEE EST PERTINENTE PLUS ELLE SE RAPPROCHE DU ROUGE SUR LE SCHEMA

Parmi ces trois formules, la mesure d'entropie (entropy sampling) est la technique la plus couramment utilisée.

On verra par la suite qu'un des grands désavantages de la technique de sélection par incertitude est la tendance qu'a le modèle à sélectionner des « outliers » et non pas des données représentatives de la répartition générale. De ce fait, les techniques de « Diversity Sampling » (6.2.7) ou de sélection dite hybride (6.3.2) font peu à peu leur apparition et tendent à sélectionner des données plus représentatives.

6.2.3. QUERY BY COMMITTEE

Le principe de la stratégie « *Query by committee* » (Requête par votes) est l'utilisation de plusieurs modèles, chacun entraîné sur l'ensemble des données labélisées, pour déterminer les données pertinentes à étiqueter.
L'ensemble des modèles utilisés forme ce que l'on appelle un « *committee* ». En fonction de la prédiction réalisée par chaque modèle sur l'ensemble des données non étiquetées, on va sélectionner celles dont les modèles sont le plus en désaccord.
Pour sélectionner les différents modèles qui tenteront de déterminer la donnée à labéliser, on peut :

- Initialiser les paramètres de chaque modèle de façon aléatoire
- Partitionner les dimensions de l'espace des données
- Utiliser le principe de *Bagging* [2]
- Utiliser le principe de *Boosting* [2]

Les performances constatées ne permettent pas de choisir la technique ni le nombre de modèle à utiliser car elles dépendent du jeu de données.

Il existe plusieurs méthodes permettant de calculer le désaccord entre les modèles. Les plus couramment utilisées sont :

- Le calcul d'entropie :

Équation 4 : Calcul d'entropie pour la stratégie de sélection par committee

$$x^* = argmax_x\left(\frac{1}{C} - \sum_i^Y \frac{V(y_i)}{C}\log\frac{\mathrm{V(y_i)}}{C}\right)$$

Avec C la taille du committee et $\mathrm{V(y_i)}$ *est le nombre de votes attribués à l'étiquette yi*

- La moyenne de Kullback-Leibler (KL)

Équation 5 : Moyenne de Kullback-Leibler

$$x^* = argmax_x\left(\frac{1}{C} - \sum_{c=1}^{C}\sum_i P_\theta(\mathrm{y_i}|x)\log\left(\frac{P_{\theta_c}(y_i|x)}{P_C(y_i|x)}\right)\right)$$

Des algorithmes de sélection par comité plus élaborés tels que « *Active-Decorate* » a des performances comparables aux techniques de bagging et boosting. [2]

Dans les problèmes de régression, une mesure de la variance peut aussi être utilisée dans le calcul du désaccord.

6.2.4. EXPECTED MODEL CHANGE

L'objectif de la stratégie élaborée par Nicholas Roy et Andrew McCallum, [3], est de mesurer le changement apporté par la labélisation de la donnée. Celle qui maximise ce changement sera celle choisie pour la labélisation. Pour ce faire, la mesure de la norme du gradient attendue (« *expected gradient length* ») permettra d'approximer la valeur absolue du changement apporté. Cette technique peut être appliquée sur tous les modèles se basant sur un « *gradient training* ».
Le principe repose sur la supposition que la donnée induisant un très gros changement aux paramètres du model lors de l'entrainement est une donnée qui mérite d'être labélisée. Or l'optimisation des paramètres est réalisée par une descente de gradient, il faut donc sélectionner la donnée qui maximise la norme du gradient.

Lors d'une descente de gradient on a :

Équation 6 : Descente de gradient

$$\theta \coloneqq \theta - \alpha\frac{\partial l_{x_i}(\theta)}{\partial\theta}$$

avec alpha le learning rate et lxi(theta) la fonction de coût

La formule donnant le vecteur à labéliser est la suivante :

Équation 7 : Maximisation du gradient

$$x^* = argmax_x \sum_i P_\theta(y_i|x)\|\nabla l_\theta(\langle x, yu_i\rangle)\|$$

Avec $\nabla l_\theta(\langle x, yu_i\rangle)$ le gradient de la fonction l selon le modèle théta obtenu en ajoutant le couple $< x, y_i >$ à l'ensemble des données labélisées.

Cette méthode est couteuse en termes de complexité en temps de calcul et n'est pas efficace lorsque la magnitude d'une des dimensions est très supérieure aux autres. Chen et Rosenfeld, par leur méthode de régularisation des paramètres, ont montré que l'on peut réduire les effet du problème de dimensionnement [5].

Cette méthode peut être utilisée pour les problèmes de régression [6].

6.2.5. EXPECTED ERROR REDUCTION

L'objectif de cette stratégie est de réduire l'erreur de généralisation en ajoutant le couple <x,y> à l'ensemble des données labélisées. On va entrainer le modèle sur l'ensemble des données labélisées, le couple <x,y> compris, et calculer l'erreur faite sur la prédiction sur l'ensemble des données non labélisées. Ne connaissant pas le label des données sur lesquelles on va tester notre modèle, on évalue en supposant que chaque donnée appartienne à chaque label possible.
On va donc choisir une mesure de calcul d'erreur tel que le *Binary Loss*.
On en vient donc à la formule :

Équation 8 : Minimisation de la fonction de cout 0/1

$$x^*_{0/1} = argmin_x \sum_i P_\theta(y_i|x)(\sum_{u=1}^{U} 1 - P_{\theta^{+(x,y_i)}}(\hat{y}|\mathrm{x}^u))$$

Avec $\theta^{+(x,y_i)}$ le modèle entrainé en utilisant le couple $< x, y_i >$.

En utilisant *Log-Loss* pour la fonction de coût :

Équation 9 : Minimisation de la fonction de coût log-loss

$$x^*_{log} = argmin_x \sum_i P_\theta(y_i|x)(-\sum_{u=1}^{U}\sum_i P_{\theta^{+(x,y_i)}}(\hat{y}|\mathrm{x}^u)\log\left(P_{\theta^{+(x,y_i)}}(\hat{y}|\mathrm{x}^u)\right))$$

En générale c'est une stratégie très efficace mais inutilisable dans la plupart des cas car demandant une puissance de calcul extrêmement élevée. En effet, non seulement il faut prédire l'erreur d'ajout sur l'ensemble des données non labélisées à chaque requête mais surtout recréer un modèle que l'on réentraîne incrémentalement pour chaque labélisation possible.

6.2.6. MAXIMISATION DE LA VARIANCE

La maximisation de la variance attendue, [4], est une autre stratégie pour optimiser le choix des données à labéliser. Le principe est de prévoir la nouvelle variance à la suite de la labélisation d'une donnée x et maximiser celle-ci.

Pour un problème de classification, on va tout d'abord, pour tout exemple x_j non labélisé, entrainer le modèle sur l'ensemble de donnée non labélisées en y ajoutant le couple <xj,yi> puis calculer l'entropie obtenue lors de la labélisation de la donnée xj pour chaque label yi :

$$e_j = \sum_i P(y_i|x_j) log P(y_i|x_j)$$

Après avoir obtenu la variance attendue pour chaque couple possible, on sélectionnera la donner qui maximisera cette variance.
Pour les problèmes de régression on peut utiliser l'information de Fisher afin de calculer la variance attendue.

Cette technique prend en compte la représentativité de la donnée.

6.2.7. DENSITY WEIGHTED REDUCTION

La méthode de « Densité Pondérée » est utilisée pour sélectionner des données représentatives de la répartition globale. La supposition faîte ici est qu'une région marginale peut être moins pertinente qu'une région dense en donnée.
On peut donc aller récupérer les données qui sont dans une zone « dense en information » en utilisant la densité par similarité.

Équation 10 : Equation de la sélection par densité

$$x^* = argmax_x(\phi_A(x) * (\frac{1}{U}\sum_{u=1}^{U} sim(x, x^u))^{\beta}$$

On multiplie la moyenne de similarité avec la valeur d'information du vecteur x à labéliser $\phi_A(x)$.
Bien sûr, toutes les techniques de calcul de densité peuvent être utilisées dans cette stratégie.

Figure 3 : ILLUSTRAION DE LA DIFFERENCE DES STRATEGIE DE DIVERSITY SAMPLING ET D'UNCERTAINTY SAMPLING AVEC LE POINT A SELECTIONNE PAR LA METHODE UNCERTANTY ALORS QUE LE POINT B LE SERAIT PAR LA METHODE DE DIVERSITY.

L'algorithme ALEC [5] réalise de l'Active Learning par clustering en densité.

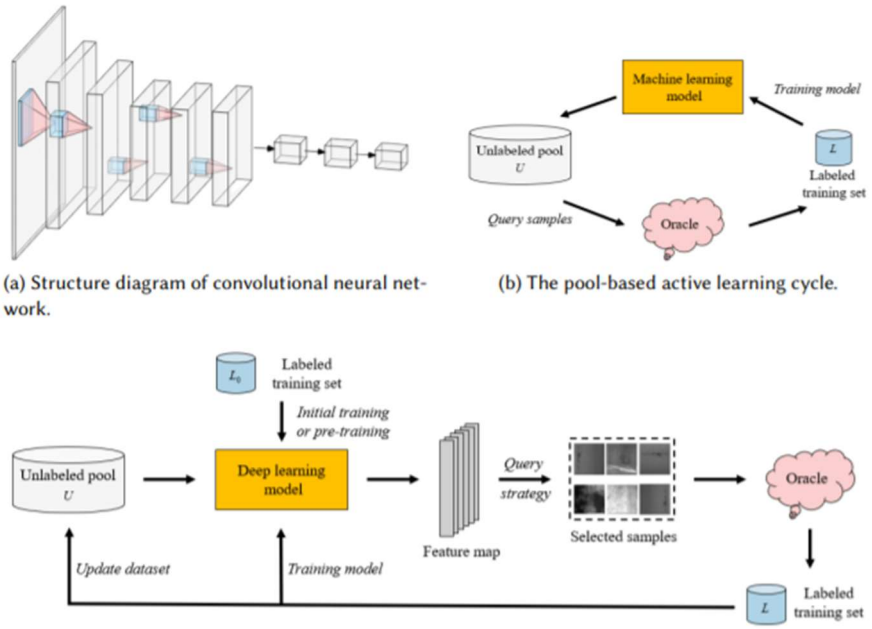
6.2.8. TABLEAU COMPARATIF

Tableau 1 : Liste des différentes méthodes d'active learning classique et énumération des avantages et inconvénients

Query Strategy	Forces	Faiblesses
Uncertainty Sampling	• Mise en place simple • Diversité des techniques de calcul d'incertitude	• Se concentre beaucoup sur les outliers • Mène à des problèmes d'overfitting lorsque le budget est bas
Query by committee	• Flexibilité sur le choix des modèles formant le committee • Diversité des techniques de mesure du désaccord	• L'optimisation du choix du committee et de la mesure de désaccord peut être fastidieux
Expected Model Change	• Fonctionne sur tout algorithme utilisant du gradient training	• Complexité importante • Mauvaise représentativité de la donnée
Expected Error Reduction	• Applicable à tout type d'algorithme	• Complexité extrêmement importante
Variance Reduction	• Utile pour les problèmes de régression • Maintient la représentativité de la donnée	• Moins utile pour les problèmes de classification
Density Weighted Methods	• Résout le problème de représentativité des données • Réduit les risques d'overfitting • Résultats prometteurs • Flexibilité sur le choix des méthodes de calcul de densité et de calcul de valeur informative • On peut ici réutiliser le modèle de détection de nouveauté	

6.3. DEEP ACTIVE LEARNING

L'émergence du Deep Learning a permis de renforcer l'intérêt des data scientist pour l'Active Learning. En effet le Deep Learning est gourmand en données car elles lui permettent d'optimiser un grand nombre de paramètres nécessaires pour obtenir des résultats pertinents.
Dans ce chapitre nous allons voir comment le Deep Active Learning peut permettre de minimiser la quantité de données labélisées tout en maintenant la puissante capacité d'apprentissage des algorithmes de Deep Learning.
L'Active Learning classique gère difficilement les données de grandes dimensions, c'est donc l'enjeux majeur du DeepAL.



(a) Structure diagram of convolutional neural network.

(b) The pool-based active learning cycle.

Figure 4 : DIAGRAMME PRESENTANT LA METHODE DE DEEP ACTIVE LEARNING APPLIQUE A UN RESEAU DE NEURONES DE CONVOLUTION. SOURCE : (REN, CHANG, HUANG, LI, GUPTA, CHEN & WANG, 2022)

6.3.1. PROBLEMATIQUES

L'Active Learning rencontre de nombreuses problématiques lors de son application en Deep Learning.

- L'incertitude en Deep Learning est une variable floue et difficile à déceler. Bien que l'on puisse obtenir une probabilité de distribution d'une donnée non labélisée grâce à des couches telles que l'ajout d'une « *softmax layer* », cette prédiction s'avère en pratique inutilisable car bien trop confiante (probabilité proche de 1 pour des données mal classifiées). En pratique, l'utilisation de cette méthode s'avère même moins efficace qu'une sélection aléatoire.

- Il est nécessaire d'avoir bien plus de donnée pour entraîner un modèle de Deep Learning qu'un modèle de Machine Learning classique, or l'Active Learning appliqué au Machine Learning réduit drastiquement le nombre de donnée d'entrainement.

- La procédure d'entraînement d'un modèle de Deep Learning et celle de l'Active Learning sont incompatibles utilisées telles quelles. L'Active Learning entraîne le modèle indépendamment de l'algorithme de sélection de la donnée utilisée or en Deep Learning ces deux procédures sont activement corrélées (optimisation simultanée durant l'entraînement).

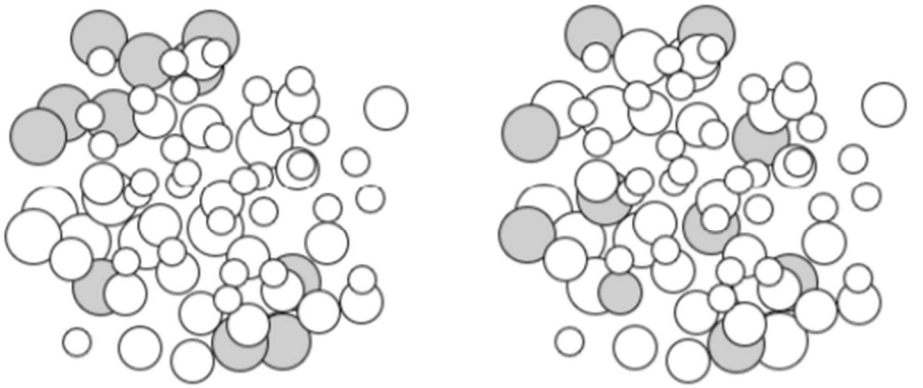
Dans ce qui suit nous verrons l'état de l'art du Deep Active Learning ainsi que les solutions trouvées aux problèmes précédemment cités.

6.3.2. OPTIMISATION DES « QUERY STRATEGIES » POUR L'APPLICATION AU DEEPAL

Dans le premier chapitre nous avons pu définir les différentes stratégies de sélection de requête pour de l'Active Learning classique. Ces méthodes doivent pour la plupart être réadaptées pour convenir aux algorithmes de Deep Learning.

- **Méthode de sélection par Batch-Mode**

La méthode de sélection par Batch-Mode (Batch-Mode query strategy) est celle qui paraît la plus intuitive quand on veut travailler sur l'entraînement d'un modèle de Deep Learning et c'est aussi la première solution pour l'optimisation des QS pour le DeepAL. C'est une méthode qui permet de sélectionner un ensemble de donnée à labéliser par requête. A l'inverse de la labélisation en série elle permet de paralléliser les tâches de labélisation (ce qui peut être utile quand on peut disposer de plusieurs Oracles) et c'est une technique qui peut être applicable au Deep Learning. En effet la labélisation en série, parfois utilisée en Active Learning classique, est inefficace pour les modèles de Deep Learning et mène régulièrement à du surapprentissage.
Une méthode naïve serait de sélectionner le sous-groupe de données à labéliser en maintenant la technique de labélisation en série et en itérant k fois. On récupère ainsi les k plus pertinentes données selon la QS utilisée.
En pratique cette méthode n'est pas efficace car on sélectionne un ensemble composé de données beaucoup trop similaires. La solution serait de considérer la corrélation entre les points sélectionnés et tendre à minimiser celle-ci.



(a) Batch query strategy considering only the amount of information.

(b) Batch query strategy considering both information volume and diversity.

Figure 5 : DIFFERENCE ENTRE LA SELECTION PAR BATCH MODE UTILISANT L'UNCERTAINTY SAMPLING ET LA SELECTION PAR BATCH MODE INCLUANT LE DIVERSITY SAMPLING. ADAPTEE DE : (REN, CHANG, HUANG, LI, GUPTA, CHEN & WANG, 2022)

La stratégie établie par Cardoso and all nommée « *Ranked batch-mode active learning* », [5], évalue l'importance de la donnée par :

Équation 11 : Score du rank batch mode

$$score = \alpha\left(1 - \Phi(x, X_{labeled})\right) + (1 - \alpha)U(x)$$

Où $\alpha = \frac{|x_{labeld}|}{|x_{labeled}|+|x_{unlabeled}|}$, U(x) est l'incertitude obtenue à partir d'une méthode de « *uncertainity sampling* » et $\Phi(x, X_{labeled})$ est une mesure de similarité par exemple en utilisant la similarité cosine.
Ici la fonction Φ permet de garder une cohésion dans la donnée sélectionnée avec la donnée réelle.

On peut retrouver dans la littérature d'autres méthodes de sélection par *batch mode telle que batchBALD* [6].

- **Stratégie de sélection de requête hybride**

Il est relativement simple de mesurer l'incertitude liée à une donnée pour un algorithme de Machine Learning classique. Cependant, comme précisé précédemment, ce n'est pas le cas pour un algorithme de Deep Learning. On va donc utiliser une stratégie de sélection de requête hybride (HQS pour Hybrid Query Selection) sur un sous-groupe de donnée (batch-mode).
Cette stratégie va se baser sur des scores de similarité calculés entre les différents vecteurs du sous-groupe. Une technique utilisée dans : « Deep Similarity-Based Batch Mode Active Learning with Exploration-Exploitation » propose les formules de sélection suivantes :

Équation 12 : Redondance

$$R(S) = \sum_{x_i \in S} \sum_{x_j \in S} Sim(x_i, x_j), \quad Sim\left(x_i, x_j\right) = f(x_i) M f(x_j)$$

Équation 13 : Densité par coefficient de redondance

$$I(S) = E(S) - \frac{\alpha}{|S|}\mathrm{R(S)}$$

Où R(S) représente la redondance dans le sous-groupe S. f(x) représente l'application du modèle de Deep Learning f au vecteur x. M est la matrice de similarité. α est un coefficient permettant de faire varier l'importance de la partie « ressemblance » vis-à-vis de celle « d'information ».
Bien que cette technique soit souvent performante, il existe des cas où le jeu de données n'est pas approprié à des techniques de *diversity sampling*.
D'autres techniques telles que le BADGE (*Batch Active Learning by Diverse Gradient Embeddings* [8]) permettent de gérer l'importance donnée à la partie *Uncertainity* et à la partie Diversity de façon automatique sans la nécessité de configurer manuellement les hyperparamètres.

La technique de sélection nommée WAAL (Wasserstein Adversarial Active Learning [9] et [10]) utilise la dérivation de la fonction de coût de la distance Wasserstein pour établir une corrélation entre les sous-groupes du batch mode.
Enfin, la méthode TA-VAAL (pour Task-Aware Variational Adversarial Active Learning) permet aussi de combiner diversity sampling et uncertainity sampling de façon dynamique et obtient très souvent des performances remarquables sur tout type de jeu de données.

Bien que ces différentes techniques hybrides de sélection de requête aient des performances supérieures aux seules techniques de *Uncertainty Sampling*, ce sont ces dernières qui sont le plus couramment utilisées en DeepAL car elles sont plus simples à mettre en place puisqu'elles sont compatibles avec la fonction softmax en sortie des algorithmes de Deep Learning.

6.3.3. DEEP BAYESIAN ACTIVE LEARNING (DBAL)

Pour obtenir des résultats traitables par les méthodes de *Uncertainty Sampling*, des modèles de *Deep Bayesian Active Learning* (DBAL) combinant *Deep Learning* et *Active Learning* ont été mis en place par Yarin Gal dans sa thèse sur la recherche d'incertitude pour le DL [11]. Un réseau de neurones dit Bayésien prend en entrée le vecteur x et fournit le vecteur y qui est une distribution des poids de prédictions obtenues pour le vecteur x sur chacune des classes yi. Ainsi, en sortie on a un vecteur y dont la cellule yi est la probabilité que le vecteur x soit labélisé par le label yi : p(yi|x).

On cherche à obtenir ces trois types de mesure d'incertitude :

- *MaxEntropy* (Shannon, 2001). The higher the entropy of the predictive distribution, the more uncertain the model is:

$$H[y|\boldsymbol{x},\boldsymbol{\theta}] = -\sum_{c} p(y=c|\boldsymbol{x},\boldsymbol{\theta})\log p(y=c|\boldsymbol{x},\boldsymbol{\theta}) \quad (3)$$

- *Bayesian Active Learning by Disagreement (BALD)* (Houlsby et al., 2011). Based on the mutual information between the input data and posterior, and quantifies the information gain about the model parameters if the correct label would be provided.

$$I(y,\boldsymbol{\theta}|\boldsymbol{x},\boldsymbol{\theta}) = H[y|\boldsymbol{x};\mathbb{X},\mathbb{Y}] - \mathbb{E}_{\boldsymbol{\theta}\sim p(\boldsymbol{\theta}|\mathbb{X},\mathbb{Y})}\left[H[\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\theta}]\right] \quad (4)$$

- *Variation Ratio* (Freeman, 1965). Measures the statistical dispersion of a categorical variable, with larger values indicating higher uncertainty:

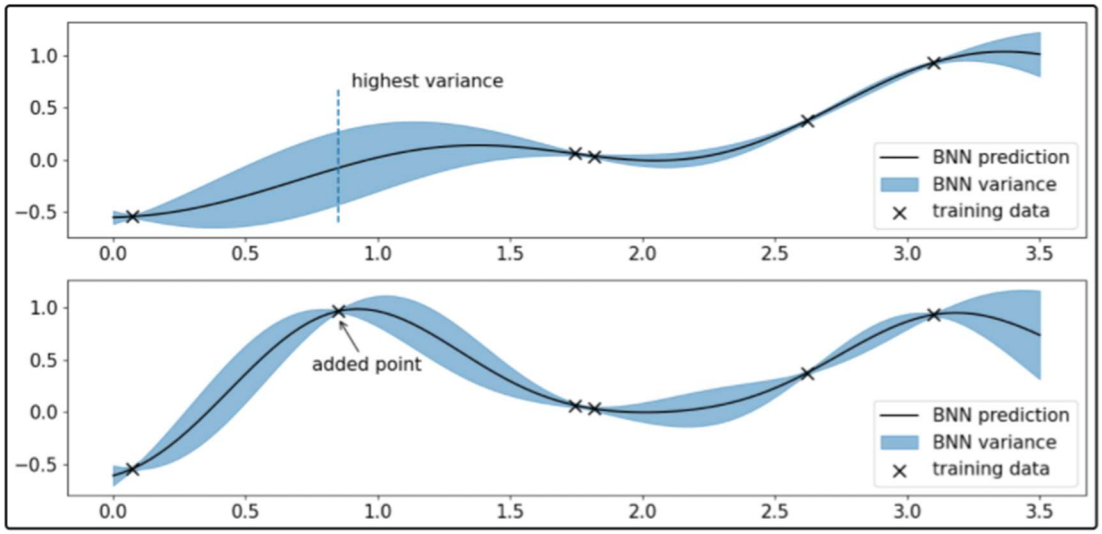
$$\text{VarRatio}(\boldsymbol{x}) = 1 - \max_{y} p(y|\boldsymbol{x},\boldsymbol{\theta}) \quad (5)$$

Ici « c » représente une classe.
Avec :

Équation 14 : Probabilité de prédiction pour BNN

$$p(y^*|x^*,X,Y) = \int p(y^*|x,\theta)p(\theta|X,Y)d\theta = E_{\theta\sim p(\theta|X,Y)}[f(x,\theta)]$$

Un réseau de neurones Bayésien permet d'approximer un « *Gaussian Process* » qui est définit par la moyenne et la covariance du réseau. Pour un réseau de neurones non Bayésien, on utilise une méthode qui s'appelle « *sampling the dropout* » démontré comme étant l'équivalent du « *Gaussian Process* » [12].
La variance obtenue (*posterior variance*) est la définition de l'incertitude recherchée. L'image ci-dessous illustre ce procédé :



d

Figure 6 : ILLUSTRATION DE L'INTERET DE CONSIDERER LA VARIANCE D'UN RESEAU DE NEURONNE BAYESIEN POUR LA LABELISATION DE LA DONNEE.

La méthode permettant d'approximer le « *Gaussain Process* » par le « *sampling dropout* » est la suivante :
On applique la méthode de Monte Carlo pour approximer l'espérance de la fonction *softmax* f du vecteur x suivant l'espace des paramètres θ de longueur K = nombre total de MC-Dropout.
On obtient :

$$E_{\theta \sim p(\theta|X,Y)}[f(x,\theta)] = \frac{1}{K}\sum_{k} p(y = c|x,\theta_k)$$

La prédiction obtenue sur la classe de x est beaucoup moins confiante. On peut ainsi appliquer les traditionnelles « *Uncertainty query strategies* ».
Néanmoins les méthode de DBAL classique s'adapte difficilement aux jeu de données volumineux. Nous allons explorer des méthodes perfectionnant celle de DBAL :

- DEBAL (Deep Ensemble Bayesian Active Learning) [13]

Le problème que tente de résoudre le DEBAL est celui de l'excès de confiance du modèle de MC-Dropout dans sa prédiction. La solution proposée est de remplacer le seul MC-Dropout par un ensemble de MC-Dropout avec chaque membre de l'ensemble initialisé de façon différente avec l'intuition que de telle manière, on couvrira une plus grande partie de l'ensemble de distribution des données.

- BatchBALD [7]

La méthode *BatchBALD* est sensiblement identique à celle de BALD à l'exception qu'elle va se concentrer sur le calcule d'incertitude entre les différents sous-groupes (batch) plutôt qu'entre les données considérées unes à unes.

Algorithm 1: Greedy BatchBALD $1 - 1/e$-approximate algorithm

Input: acquisition size b, unlabelled dataset $\mathcal{D}_{\text{pool}}$, model parameters $\mathrm{p}(\boldsymbol{\omega} \mid \mathcal{D}_{\text{train}})$

$A_0 \leftarrow \emptyset$
for $n \leftarrow 1$ **to** b **do**
  foreach $\boldsymbol{x} \in \mathcal{D}_{pool} \setminus A_{n-1}$ **do** $s_{\boldsymbol{x}} \leftarrow a_{\texttt{BatchBALD}}\left(A_{n-1} \cup \{\boldsymbol{x}\}, \mathrm{p}(\boldsymbol{\omega} \mid \mathcal{D}_{\text{train}})\right)$
  $\boldsymbol{x}_n \leftarrow \underset{\boldsymbol{x} \in \mathcal{D}_{\text{pool}} \setminus A_{n-1}}{\arg\max}\ s_{\boldsymbol{x}}$
  $A_n \leftarrow A_{n-1} \cup \{\boldsymbol{x}_n\}$
end

Output: acquisition batch $A_n = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_b\}$

- ACS-FW (Active Bayesian CoreSets with Frank-Wolfe optimization) [14]

La méthode utilisée tente d'approximer au mieux le logarithme du *posterior* du batch sélectionné pour qu'il soit relativement proche du logarithme du *posterior* du dataset complet. L'algorithme de Frank-Wolf est ensuite appliqué afin de résoudre le problème d'approximation qui en découle.
ACS-FW permet une meilleure diversification des « batch » sélectionnés comparé aux autres techniques.

- DPEs (Deep Probabilistic Ensembles) [15]

DPEs utilise la technique de « *variational inference* » pour entrainer les BNNs.
L'algorithme est décrit ci-dessous :

Algorithm 1 Active learning with DPEs.

$i \leftarrow 0$
Randomly sample b points $\{\mathbf{x}^j\}_{j=1}^b$ from unlabeled set $U^{(0)}$
Annotate this data, $\{y^j = \mathcal{A}(\mathbf{x}^j)\}_{j=1}^b$
Move $\{(\mathbf{x}^j, y^j)\}_{j=1}^b$ to initial labeled set $L^{(0)}$
total_added_samples $\leftarrow b$
while total_added_samples $\leq B$ **do**
  while model_not_converged **do** ▷ **stage (i)**
    Forward pass a mini-batch from $L^{(i)}$, $\hat{y} = \mathcal{M}^{(i)}(x)$
    Compute gradients δ of loss $l(y, \hat{y}) + \beta\Omega$
    Update DPE, $\mathcal{M}^{(i)} \leftarrow \mathcal{M}^{(i)} + \delta$
  end while
  for num_iteration_samples = 1 to b **do** ▷ **stage (ii)**
    for $\mathbf{x} \in U^{(i)}$ **do**
      $\alpha(\mathbf{x}) \leftarrow H_{ens}(\mathbf{x})$
    end for
    $choice = \arg\max_{\mathbf{x}} \alpha(\mathbf{x})$
    $y^{choice} = \mathcal{A}(\mathbf{x}^{choice})$
    Move $(\mathbf{x}^{choice}, y^{choice})$ from $U^{(i)}$ to $L^{(i+1)}$
    total_added_samples += 1
  end for
  Update b
  i += 1
end while

- ActiveLinks (Deep Active Learning for Link Prediction in Knowledge Graphs) [16]

ActiveLinks se base sur la théorie des graphs pour définir la valeur informative d'une donnée non labélisée. En plus de la mesure d'incertitude donnée par la formule de l'entropie de Shannon, ActiveLinks définit la valeur informative d'une donnée en fonction du nombre de liens qu'elle a avec le reste des données déjà labélisées. De plus, plus une donnée est liée à d'autres (indépendamment des données labélisées), plus elle sera considérée comme informative.

Aussi ActiveLinks se base sur les techniques de metaLearning qui permettent de réaliser un training des modèles de DL itérativement sur chaque requête réalisée sans pour autant refaire entièrement l'entrainement des modèles depuis zéro.

6.3.4. DENSITY BASED METHODS

La méthode de sélection par densité peut aussi être adaptée pour le DeepAL.

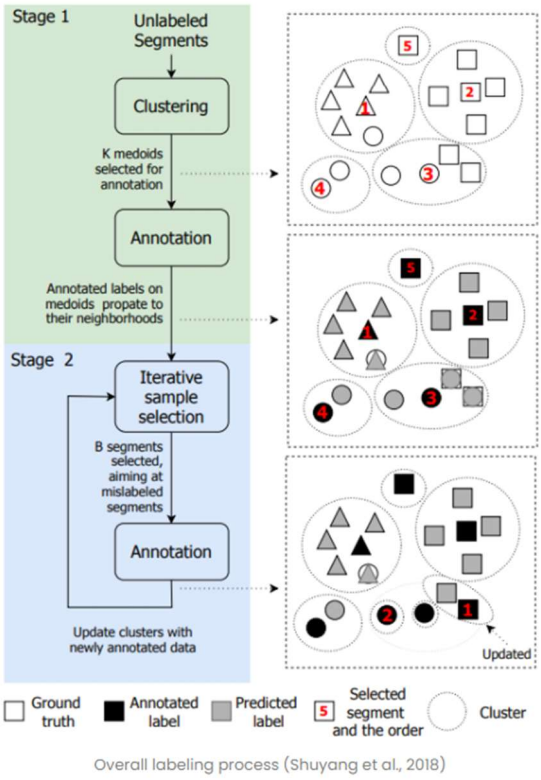


Figure 7 : DIAGRAMME REPRESENTANT LE PROCESSUS DE LABELISATION DANS UNE TECHNIQUE BASEE SUR LA SELECTION PAR DENSITE

Ci-dessous quelques algorithmes de sélection par densité qui peuvent être utilisés dans le deepAL :

- FF-Active (Farthest First Active Learning) [17] qui réalise un clustering du jeu de données non labélisé et qui détermine le centre de ces clusters en appliquant la méthode de *k-medoids*. On annote certaines données qui vont permettre à entrainer un modèle de classification comme un algorithme des k plus proche voisins ou un modèle de régression logistique. Si les données prédites sont éloignées de la classification réalisée par les modèles, alors on les ajoute dans un tableau de données à potentiellement labéliser. A partir de ce tableau, on va labéliser ceux qui sont le plus éloignés des centroïdes des clusters.

- Discriminative Active Learning [18]

L'objectif de cette méthode est de fournir un jeu de données labélisé qui est au maximum ressemblant au jeu de données non labélisé notamment en termes de distribution de la donnée. Le principe serait de regarder la probabilité qu'une donnée non labélisée appartienne à l'ensemble des données non labélisées. Si elle est élevée, on doit rajouter cette donnée aux données à labéliser (la valeur informative de la donnée est élevée). Bien que l'implémentation de cette méthode soit simple, elle ne donne pas des résultats convaincants en pratique.

6.4. AVANT DE TRAITER UN PROBLEME D'ACTIVE LEARNING

Voici une liste de questions à se poser avant de traiter un problème d'Active Learning :

- ***Quelle est la proportion des données non labélisées face à celles labélisées ?***
  Si toutes les données sont labélisées, le problème de l'Active Learning ne se pose pas. Au contraire, si la proportion de données labélisées est faible face à celle des données non labélisées alors l'Active Learning peut jouer un rôle majeur dans l'amélioration des processus d'apprentissage.

- ***Est-il couteux de labéliser des données ?***
  S'il n'est pas coûteux de labéliser les données, comme ci-dessus le problème d'AL ne se pose pas. Ensuite, il est important de bien définir le coût de la labélisation de la donnée. Est-ce un coût humain ? Est-ce un coût en temps ? En mémoire ?
  Aussi, la plupart des algorithmes d'AL suppose que le coût de labélisation est égal entre chaque donnée. En pratique, il se peut qu'une donnée soit plus simple à labéliser qu'une autre. Si l'on veut optimiser les algorithmes d'AL, il faudrait leur permettre d'intégrer le paramètre de variation de coût.

- ***En fonction de ce coût, comment déterminer le budget qui serait alloué à la labélisation des données ?***
  Le « stopping criteria » qui permet aux algorithmes d'Active Learning de savoir à quel moment la labélisation est plus couteuse qu'utile nécessite d'être définit en fonction du budget alloué à la labélisation. Ce budget peut être exprimé en termes de quantité de requête autorisée mais il peut aussi être définit en termes de performance à atteindre. Aussi, les recherches actuelles font état de nouvelles techniques de définition automatique de budget.

- ***A quel point a-t-on confiance dans les classifications réalisées par l'Oracle ?***
  Un Oracle peut réaliser des labélisations plus ou moins exactes. En effet, en fonction du type d'Oracle (expert métier, étiqueteur algorithmique, non experts …), la confiance en la labélisation réalisée peut varier. Cette confiance en l'Oracle peut impacter le choix de la méthode d'Active Learning à utiliser.

- ***Peut-on se permettre de réaliser une augmentation des données ?***
  En fonction de la donnée et de sa sensibilité, il est plus ou moins faisable de réaliser une data augmentation. Celle-ci peut être fait manuellement ou intégrée directement dans les algorithmes d'Active Learning. En générale, la combinaison d'Active Learning et de data augmentation permet d'améliorer grandement les résultats notamment dans le contexte du Deep Learning. Aussi, la data augmentation peut permettre de réduire le budget alloué à la labélisation de la donnée.

- ***Quel est le modèle de prédiction utilisé ?***
  Comme on a pu le voir précédemment, le modèle influence énormément le choix de la stratégie d'Active Learning. Ainsi, un problème de classification ne sera pas traité de la même manière qu'un problème de régression. Il en est de même entre un problème de Machine Learning traditionnel et un de Deep Learning.

- ***Est-on en présence de nombreux outliers ? Quelle importance donne-t-on à la détection d'outliers ?***
  Les outliers peuvent impacter grandement les résultats de l'Active Learning. Les stratégies de sélection par incertitude notamment déterminent les données à labéliser en fonction de sa valeur informative qui est très souvent mesurée par sa distance avec l'ensemble des données labélisées. Or c'est entre autres ce qui définit un outlier. On risque donc de se retrouver avec un modèle entrainé en majorité sur des outliers plutôt que des données représentatives. Il est donc primordial de définir l'importance donnée à la détection d'outliers plutôt qu'à la représentativité de la donnée.

6.5. EVALUER UN MODELE D'ACTIVE LEARNING

On peut se baser sur plusieurs critères pour évaluer un modèle d'Active Learning. Voici une sélection de critère permettant de juger les performances d'un de ces types d'algorithmes :

- **Accuracy finale du modèle**

  Il s'agit souvent du premier critère de jugement d'un algorithme de ML. Le principe serait donc d'évaluer l'accuracy obtenu à la fin du processus d'Active Learning, celui-ci pouvant diverger en fonction de la QS utilisée.

- **Nombre de requête nécessaire pour atteindre une accuracy fixée ou une condition d'arrêt d'un modèle d'AL**

  En effet, si l'objectif est d'atteindre une limite dans l'accuracy (limite fixée ou limite d'amélioration de l'accuracy), on peut évaluer un algorithme d'AL en fonction de la quantité de requête nécessaire.

- **Temps de training total avant atteinte du stopping criteria**

  De la même façon que précédemment, on peut à l'inverse se baser sur le temps total de running et de labélisation obtenue à l'issue du processus d'AL.

De manière général, on pourrait rassembler les deux premiers points en utilisant une métrique qui se baserait sur un calcul d'intégral de la fonction d'accuracy durant le training.

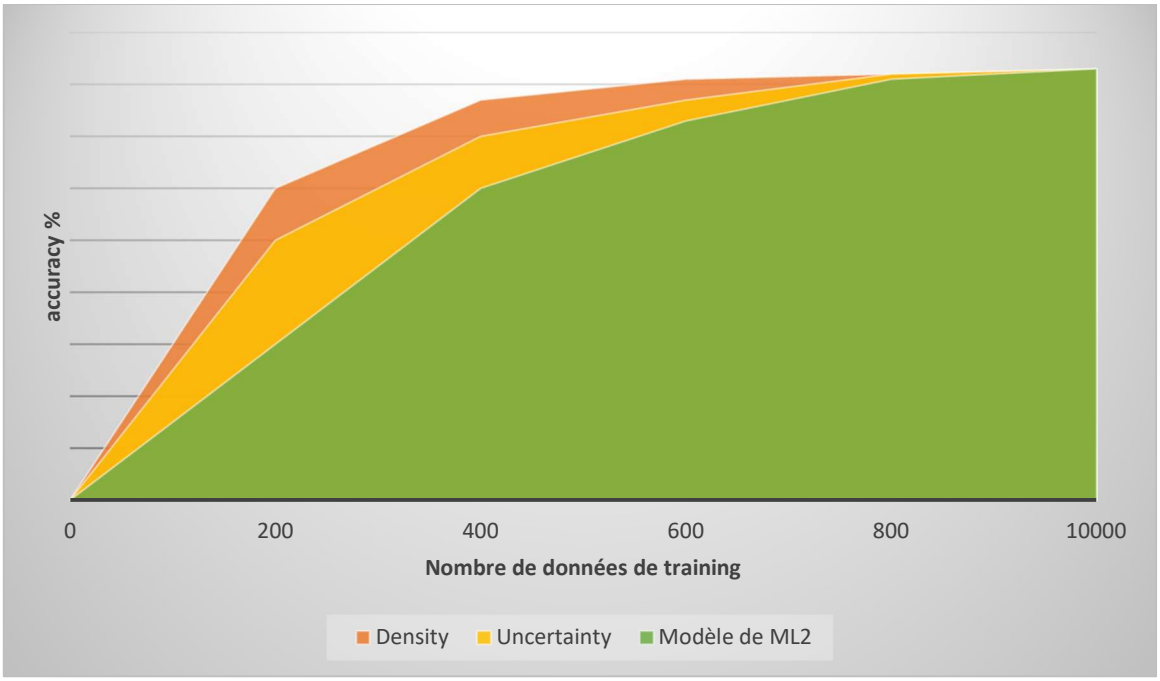


Figure 8 : : Illustration de l'accuracy obtenue sur des modèles d'Active Learning comparée à un modèle n'utilisant pas d'Active Learning. En mesurant la différence des intégrales des différentes courbes, on peut évaluer les performances des modèles d'AL sur l'accuracy en fonction de la quantité de données fournit durant la phase de training.

Aussi, les points suivant peuvent impacter les performances de l'AL. Il serait intéressant d'étudier ces différents points lorsqu'on veut évaluer un modèle d'AL :

- **Réaction du modèle en fonction de la quantité initiale de données labélisées**

  Un algorithme d'AL va plus ou moins bien se diversifier et se généraliser en partant d'un ensemble de données déjà labélisée.

- **Réaction du modèle en fonction de la quantité initiale de données non labélisées**

  Aussi la quantité totale de données non labélisée peut avoir un impact sur les performances des modèles d'AL que ce soit en termes de temps de calcul ou d'accuracy.

- **Réaction du modèle dans sa prédiction d'outliers**

  Enfin, la prédiction d'outliers est plus ou moins bien géré par les différents modèles d'AL.

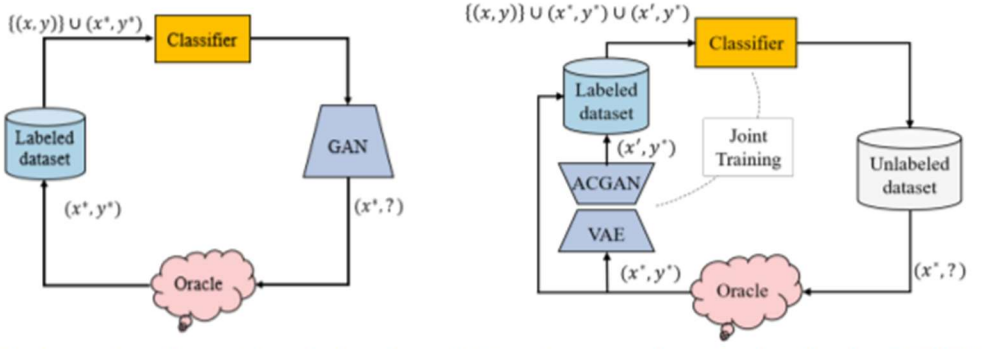
6.6. AUTRES REFLEXIONS

6.6.1. AUGMENTATION DE LA DONNEE LABELISEE

La difficulté du DeepAL est de concilier le peu de requête à l'Oracle à réaliser tout en ayant assez de données labélisées pour permettre aux algorithmes de Deep Learning de s'entraîner correctement. Pour cela il existe des techniques d'augmentation de l'espace des données labélisées sans appels à l'Oracle.

- CEAL (Cost-Effective Active Learning) permet d'augmenter l'ensemble de données d'entrainement en labélisant les données qui ont un score de prédiction élevé.

- GAAL (Generative Adversarial Active Learning [19]) propose une solution novatrice alliant data augmentation et Active Learning. Cependant cette méthode se base sur une fonction de coût qui ne doit pas être complexe en temps de calcul, or ce type de fonction de coût s'adapte mal aux problèmes d'AL. BGADL (Bayesian Generative Active Bayesian Learning) est une extension de GAAL qui optimise l'augmentation de données pour les algorithmes de DL. Ici le learner et l'optimiseur sont entraînés simultanément.



(a) Generative adversarial active learning (GAAL). (b) Bayesian generative active deep learning (BGADL).

Figure 9 : DIAGRAMME PRESENTANT LA DIFFERENCE ENTRE LES METHODES GAAL ET BGADL

On détermine une valeur informative en utilisant la méthode de Monte Carlo Dropout.

L'Oracle va par la suite labéliser cette donnée et on va générer, en faisant passer la nouvelle valeur labélisée par un encoder e() et un decodeur g(), une nouvelle valeur qui aura le même label que l'entrée.
Par un développement de Taylor au premier ordre, on peut remarquer que la valeur générée est, elle aussi, presque autant informative que la valeur d'entrée.

Le processus est illustré ci-dessous :

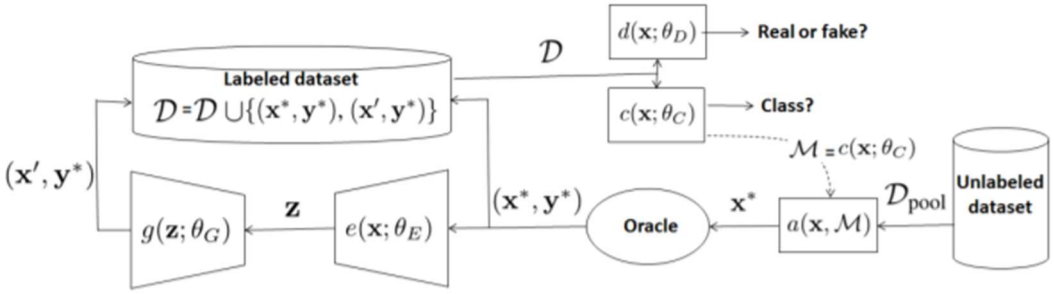


Figure 10 : SCHEMA ILLUSTRANT LE PROCESSUS DE METHODE GAAL

- VAAL (Variational Adversarial Active Learning [20])
  VAAL combine un « Variational Auto-Encoder » (VAE) qui permet une représentation de l'ensemble des données labélisées et non labélisées et un « Adversarial Network » pour calculer la probabilité qu'une donnée appartienne à l'ensemble des données labélisées. Le VAE et l'Adversarial Network sont entraînés simultanément, le VAE a pour objectif de créer une représentation favorisant la confusion sur le fait qu'une donnée appartienne à l'ensemble labélisée.

  Le process de VAAL est décrit ci-dessous :

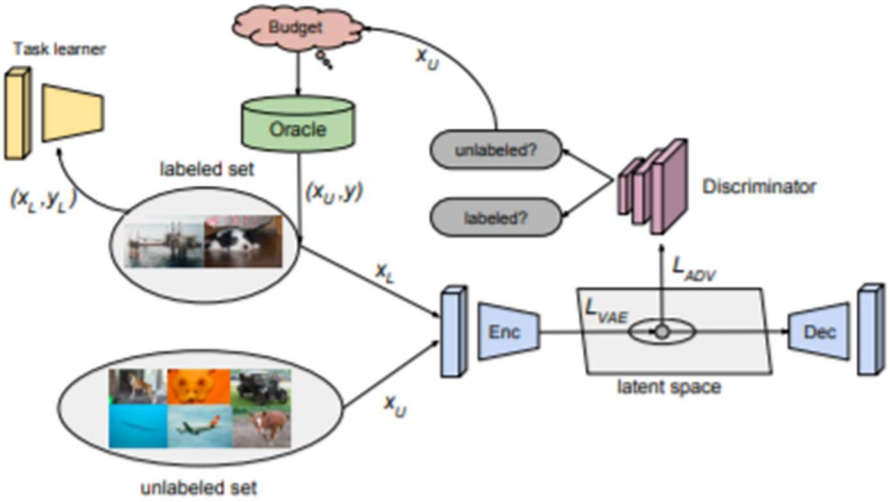


Figure 11 : DIAGRAMME ILLUSTRANT LE PROCESSUS DE LA METHODE VAAL

- ARAL (Adversarial Representation Active Learning [21])

ARAL est une extension de VAAL avec pour principal différence l'intégration des données générées et des données non labélisées dans l'entraînement des modèles de classification. En effet le decodeur de VAAL n'est pas utilisé or ici il permet de générer de nouvelles données. De plus, l'ajout d'un GAN (Generative Adversarial Network) prenant en entrée les données labélisées, non labélisées et générées permet de confirmer la sortie du Sampler de VAAL grâce au Discriminator, tout en prédisant la classe de sortie grâce au Classifier.

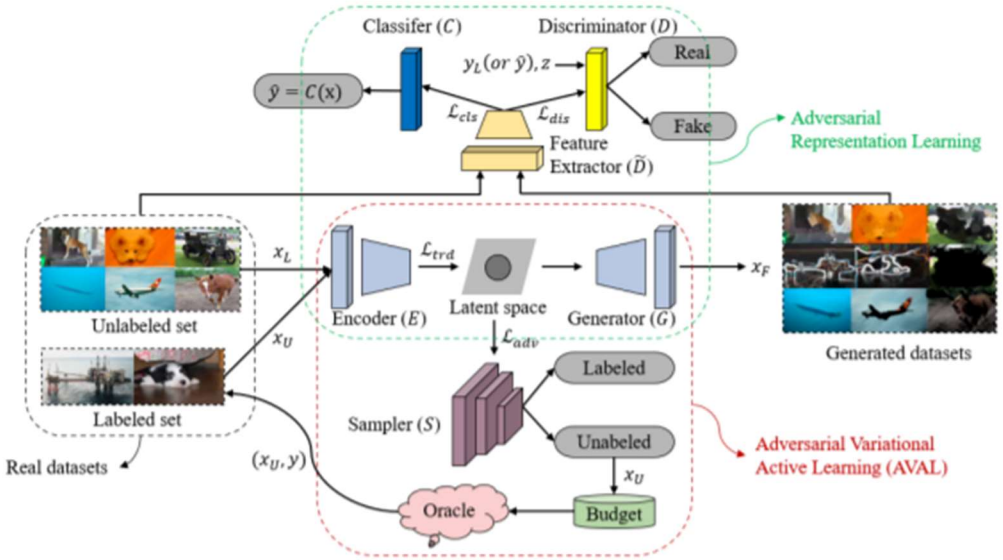


Figure 12 : DIAGRAMME ILLUSTRANT LE PROCESSUS DE LA METHODE ARAL

(Ci-dessus une illustration du procédé ARAL)

Les expérimentations ont mené à des résultats surpassant les méthodes précédemment citées.

6.6.2. L'ACTIVE LEARNING POUR LES "RANKING STRATEGIES"

Les stratégies de classement sont des algorithmes largement étudiés et utilisés dans d'autres domaines. Le développement du « World Wide Web » a amplifié la nécessité de trouver des algorithmes de classement performants. En effet, l'objectif étant qu'à partir d'une requête (une question posée à google par exemple), on obtienne des résultats classés par ordre de pertinence.
Les solutions proposées sont des algorithmes de Machine Learning qui apprennent à classer les données. Les recherches se sont principalement concentrées sur l'extraction des données importantes par document et sur le développement de ces algorithmes de ML plus que sur l'optimisation du choix des requêtes et des documents placées en entrée des algorithmes de « learning-to-rank ».
Cette technique peut être exploitée pour l'Active Learning où il y a une volonté d'obtenir un ensemble de données classées par importance dans le processus d'apprentissage des algorithmes de Machine Learning.
Les algorithmes de classements se décomposent en trois groupes :

- Pointwise approach
  Cette approche suppose que l'on peut assigner à chaque donnée du jeu d'apprentissage un score (sous format numérique) et le problème peut ainsi être cadré par un problème de régression (donner le score d'une requête).
- Pairwise approach
  Cette approche considère le problème comme une « pairwise classification » ou « pairwise regression ».
- Listwise approach

Cette approche tente d'optimiser la valeur d'une fonction de coût.

Il y a une réelle opposition entre les algorithmes d'Active Learning pour le ranking et ceux pour la classification. Ainsi, certains algorithmes d'AL de classification ne peuvent pas être adaptés au ranking notamment ceux qui utilisent des techniques de Margin.

Les algorithmes de ranking réalisent aussi bien une sélection parmi les requêtes à réaliser (Query Level) qu'une sélection parmi les données individuellement (Document Level).

Voici une liste non exhaustive d'algorithmes de ranking : RankSVM, RankBoost, GEM et ELO-DCG.

6.6.3. QUERY LEVEL ACTIVE LEARNING VS DOCUMENT LEVEL ACTIVE LEARNING

Le Query Level Active Learning détermine les requêtes les plus pertinentes à réaliser. A l'inverse, le Document Level Active Learning choisit les données les plus appropriées indépendamment de la requête.

Une étude mené par Yilmaz… (Document selection methodologies for efficient and effective learning-to-rank), montre qu'avoir plus de requêtes et moins de données est plus efficace que d'avoir peu de requêtes mais beaucoup de données.

Parfois, il est intéressant de mixer ces deux stratégies notamment dans les cas où la quantité de données non labélisées est extrêmement importante. Par exemple, dans l'objectif de classer les pages web les plus pertinentes lors d'une recherche google, on ne peut pas parcourir tous les documents du web et effectuer des opérations sur chaque document afin d'en ressortir un classement. Il serait plus judicieux avant tout de récupérer les documents les plus en liens avec la requête (Query Level) puis de sélectionner ceux qui sont les plus pertinents au sein de cette requête (Document Level).

Javed A. Asla and all propose une étude sur les stratégies de ranking combinant « Query Level » et « Document Level » [22]

Demonstration d'une méthode de ranking utilisant le Query Level et Document Level Active Learning [23]

6.6.4. REFLEXION SUR LE STOPPING CRITERIA

Il existe deux principales approches pour définir les stopping criteria lors d'une stratégie d'active learning : le « *fixed-budget approach* » et le « *dynamic-budget approach* »

Dans le premier cas, le critère déterminant la fin du processus d'Active Learning est prédéfinit. Par exemple lorsque l'on sait numériquement calculer le coût de la labélisation d'une donnée. Le stopping criteria peut alors être définit comme étant : $nombre\ de\ requête\ totales\ autorisées = \frac{coût\ maximal\ du\ processus\ d'Active\ Learning}{coût\ de\ lablisation\ d'une\ requête}$.

Le second cas est la détermination algorithmique du stopping criteria. Il présente l'avantage d'être parfois plus adapté au problème. En effet, lors d'une approche statique, on peut parfois rester dans le budget prédéfinit et tout de même donner à l'algorithme plus de données labélisées que nécessaire. Au contraire, on peut aussi rester dans le budget mais ne pas permettre à l'algorithme d'atteindre ses objectifs de performances. La supposition principale est que l'impact de la labélisation d'une donnée varie au cours du processus d'apprentissage (la millième donnée labélisée n'aura pas le même impact que la première).

Le critère permettant de stopper le processus d'apprentissage au moment opportun n'est pas seulement étudié en Active Learning. Le « early stopping » dans la littérature sur les réseaux de neurones permettant de réduire le temps de calcul et l'overfitting est un exemple d'application répandu de stopping criteria.

Hino and all propose un stopping criteria independent de la fonction de coût [24]. L'objectif est d'observer la convergence de la prédiction lors de l'apprentissage et stopper le processus d'AL lorsqu'on estime qu'il y a une réelle convergence.

6.6.5. REFLEXION SUR LA DETECTION DE NOUVEAUTE

Lorsque l'ensemble des classes n'est pas connu et qu'il existe donc des données n'appartenant à aucune des classes connues, l'Active Learning peut être mis en difficulté dans son choix de données nécessitant d'être labélisées. Par la suite, nous essaierons de poser les bases du problème et de proposer des solutions adaptées.

Le premier cas de figure est lors d'une stratégie d'Active Learning basée sur de l'Uncertainty Sampling. On peut apriori faire l'hypothèse que les données sélectionnées seront fréquemment les données qui n'appartiennent à aucune des classes. Cette stratégie peut être utile lorsqu'on recherche à découvrir un maximum de classes à défaut de prédire au mieux les classes connues. On peut néanmoins se retrouver à sélectionner les outliers des classes connues.
Dans le cas où l'on veut garder une cohérence dans la répartition des données de training sélectionnées par l'algorithme d'Active Learning par rapport à la répartition réelle, les algorithmes de sélection par densité ayant un coefficient élevé sur la partie « représentativité » peut être la solution à notre problème. Une autre solution serait de créer une classe où toutes les données inconnues y seraient répertoriées avec en contrepartie une précision diminuée.

6.6.6. QUERIES GENERATION

Dans certains cas d'Active Learning, l'algorithme peut être amené à générer des requêtes qui devraient lui permettre d'améliorer ses futures prédictions. Une recherche approfondie sur ce scénario mériterait d'être réalisé afin de compléter ce rapport sur l'état de l'art.
Huang and all, dans "Active Learning with Query Generation for Cost-Effective Text Classification" [25] , ont établi un Framework afin de générer les requêtes les plus appropriées pour un modèle de NLP.

7. RESUME ET CONCLUSION

L'objectif de ce rapport était de réaliser un compte rendu global de l'état de l'art sur les méthodes d'Active Learning et Deep Active Learning.
Nous avons pu définir tout d'abord ce qu'était l'Active Learning, son but et les différents scénarios mettant en œuvre l'Active Learning. Ensuite nous avons détaillé les différentes stratégies et les principes de fonctionnement de l'Active Learning.
Aussi, l'émergence du Deep Learning a nécessité une adaptation des méthodes d'Active Learning. On a ainsi parcouru l'ensemble des méthodes existantes du DeepAL tout en citant certaines applications concrètes, parfois des solutions innovantes proposées dernièrement. Ce domaine étant en plein essor, il est nécessaire de rester à l'écoute de l'actualité du DeepAL.
Enfin nous avons proposé une liste de question permettant de bien cadré un problème d'Active Learning avant le lancement d'un projet.

8. BIBLIOGRAPHIE

[1] M. Schuster et K. K. Paliwal, «Bidirectional Recurrent Neural Networks,» *IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 45, NO. 11,* pp. 2673-2680, 1997.

[2] N. Abe et H. Mamitsuka, «Query Learning Strategies using Boosting and Bagging,» 1998.

[3] P. Melville et R. J. Mooney, «Diverse Ensembles for Active Learning,» 2004.

[4] N. Roy et A. McCallum, «Toward Optimal Active Learning through Monte Carlo Estimation of Error Reduction».

[5] S. F. Chen et R. Rosenfield, «A Survey of Smoothing Techniques for ME Models,» 2000.

[6] W. Cai, M. Zhang et Y. Zhang, «Active Learning for ranking with sample density,» 2015.

[7] Y. Yang et M. Loog, «A Variance Maximization Criterion for Active Learning,» 2018.

[8] M. Wanga, F. Minb, Z.-H. Zhang et Y.-X. Wu, «Active Learning through density clustering,» 2017.

[9] T. N.C.Cardoso, R. M. Silva, S. Canuto, M. M. Moro et M. A. Gonçalves, «Ranked batch-mode active learning,» 2017.

[10] A. Kirsch, v. A. Joost et Y. Gal, «BatchBALD: Efficient and Diverse Batch Acquisition for Deep Bayesian Active Learning,» 2019.

[11] J. T. Ash, C. Zhang, A. Krishnamurthy, J. Langford et A. Agarwal, «Deep Batch Active Learning by Diverse, Uncertain Gradient Lower Bounds,» 2020.

[12] «Low Budget Active Learning Via Wassertein Distance: An Integer Programming Approach,» 2022.

[13] C. Shui, F. Zhou, C. Gagne et B. Wang, «Deep Active Learning: Unified and Principled Method for Query and Training,» 2020.

[14] Y. Gal, «Uncertainty in Deep Learning,» 2016.

[15] Y. Gal et Z. Ghahramami, «Dropout as a Bayesian Approximation: Representing Model Uncertainty in Deep Learning,» 2016.

[16] R. Pop et P. Fulop, «Deep Ensemble Bayesian Active Learning: Addresing the Mode Collapse Issue In Monte Carlo Dropout Via Ensembles,» 2019.

[17] R. Pinsler , J. Gordon, E. Nalisnick et J. M. Hernandez Lobato, «Bayesian Batch Active Learning as Sparse Subset Approximation,» 2019.

[18] K. Chitta, J. M. Alvarez et A. Lesnikowski, «Large Scale Visual Active Learning with Deep Probabilistic Ensembles,» 2019.

[19] N. Ostapuk, J. Yang et P. Cudré-Mauroux, «Active Link: Deep Active Learning for Link Prediction in Knowledge Graphs».

[20] S. Basu, A. Banerjee et R. J. Mooney, «Active Semi Supervision for Pairwise Constrained Clustering,» 2004.

[21] D. Gissin et S. Shalev-Shwartz, «Discriminiative Active Learning,» 2019.

[22] Goodfellow, Pouget-Abadie, Mirza, Xu, Warde-Farley, Ozair, Courville et Bengio, «Generative Adversarial Nets,» 2014.

[23] S. Sinha , S. Ebrahimi et T. Darell, «Variational Adversarial Active Learning».

[24] A. Mottaghi et S. Yeung, «Adversarial Representation Active Learning,» 2019.

[25] J. A. Aslam, A. Kanoulas, V. Pavlu, S. Savey et E. Yilmaz, «Document Selection Methodologies for Efficient and Effective Learning-to-Rank,» 2014.

[26] B. Long, O. Chapelle, Y. Zhang, Y. Chang, Z. Zheng et B. Tseng, «Active Learning for Ranking through Expected Loss Optimzation».

[27] H. Ishibashi et H. Hino, «Stopping criterion for active learning based on deterministic generalisation bounds,» 2020.

[28] Y.-F. Yan, S.-J. Huang, S. Chen, M. Liao et J. Xu, «Active Learning with Query Generation for Cost Effective Text Classification,» 2020.

[29] P. Ren, Y. Xiao, X. Chang, P. Huang, Z. Li et X. Wang, «A Survey of Deep Active Learning,» 2021.

[30] U. o. Wisconsin-Madison, «Active Learning Litterature Survey,» 2010.

[31] Kim, Park, K. I. Kim et Chun, «Task-Aware Variational Adversarial Active Learning,» 2021.

[32] T. Tran, T.-T. Do, I. Reid et G. Carneiro, «Bayesian Generative Active Deep Learning,» 2019.

[33] J. Brownlee, «A Gentle Introduction To Generative Adversarial Network (GAN's),» 2019.

[34] P. Melville et R. J. Mooney, «Diverse Ensemble for Active Learning,» 2004.

[35] C. Y. Benbin, Zhang et J. Zhou, «Maximizing Expected Model Change for Active Learning in Regression,» 2013.

[36] Y. Yang et M. Loog, «Active Learning using Uncertainty Information,» 2017.

[37] M. Li, X. Liu, J. van de Weijer et B. Raducanu, «learning to Rank for Active Learning: A Litwise Approach,» 2020.

[38] R. Pinsler, J. Gordon, E. Nalisnick et J. M. Hernández-Lobato, «Bayesian Batch Active Learning as Sparse Subset Approximation».

[39] R. Pinsler, J. Gordon et E. Nalisnick, «Bayesian Batch Active Learning as Sparse Subset Approximation,» 2019.

[40] Y.-F. Yan, S.-J. Huang, S. Chen, J. Xu et M. Liao, «Active Learning with Query Generation for Cost-Effective Text Classification,» 2020.

ANNEXE

---PAGE VIDE---