

Apprentissage par renforcement de stratégies d'apprentissage actif : une évaluation

Louis Desreumaux*, Vincent Lemaire**

*Université de Technologie de Compiègne, **Orange Labs

Résumé. L'apprentissage actif est le domaine qui regroupe l'ensemble des stratégies permettant de construire itérativement l'ensemble d'apprentissage d'un modèle en interaction avec un expert humain, l'objectif étant de sélectionner les exemples les plus informatifs pour minimiser le coût d'étiquetage. Bien que ce domaine soit assez ancien, la mise en œuvre de stratégies d'apprentissage actif en contexte industriel est encore difficile. En particulier, il n'existe pas de stratégie dans l'état de l'art qui soit meilleure que les autres dans tous les contextes. Dans cet article, nous évaluons une méthode représentative d'une orientation récente de la recherche en apprentissage actif vers des méthodes de méta-apprentissage permettant d'apprendre de nouvelles stratégies d'apprentissage actif (Konyushkova et al., 2019). Nous comparons notamment les performances d'une stratégie apprise *via* cette méthode avec celles d'une stratégie d'échantillonnage basé sur l'incertitude, lauréate de plusieurs benchmarks (Yang et Loog, 2018; Pereira-Santos et al., 2019).

1 Introduction

L'apprentissage actif regroupe l'ensemble des stratégies de sélection d'exemples permettant de construire itérativement l'ensemble d'apprentissage d'un modèle en interaction avec un expert humain, aussi appelé oracle. Il s'agit de sélectionner les exemples les plus informatifs pour minimiser le coût d'étiquetage.

Dans cet article, nous nous plaçons dans le cadre de l'échantillonnage sélectif, dans lequel les stratégies manipulent un ensemble d'exemples $\mathcal{D} = \mathcal{L} \cup \mathcal{U}$ de taille constante, où $\mathcal{L} = \{(\mathbf{x}_i, y_i)\}_{i=1}^l$ est l'ensemble des exemples étiquetés et $\mathcal{U} = \{\mathbf{x}_i\}_{i=l+1}^n$ est l'ensemble des exemples non étiquetés. Dans ce cadre, l'apprentissage actif est un processus itératif qui s'exécute jusqu'à l'épuisement d'un budget d'étiquetage ou l'atteinte d'un seuil de performance prédéfini. Chaque itération débute par la sélection de l'exemple le plus informatif $\mathbf{x}^* \in \mathcal{U}$. Cette sélection s'effectue généralement à partir des informations collectées lors des itérations précédentes (prédictions d'un classifieur, mesures de densité, etc.). L'exemple \mathbf{x}^* est alors soumis à l'oracle qui retourne la classe y^* correspondante, et le couple (\mathbf{x}^*, y^*) est incorporé dans \mathcal{L} . Le nouvel ensemble d'apprentissage est alors utilisé pour améliorer le modèle et les nouvelles prédictions sont exploitées à l'itération suivante.

Les mesures d'utilité définies par les stratégies d'apprentissage actif de la littérature (Settles, 2012) diffèrent dans leur positionnement selon un dilemme entre l'exploitation du classifieur courant et l'exploration des données d'apprentissage. La sélection d'un exemple non

étiqueté dans une région inconnue de l'espace des observations \mathbb{R}^d contribue à explorer les données, de manière à limiter le risque d'apprendre une hypothèse trop spécifique à l'ensemble \mathcal{L} courant. À l'inverse, la sélection d'un exemple dans une région échantillonnée de \mathbb{R}^d affine localement le modèle prédictif.

La lecture des résultats expérimentaux de l'état de l'art permet de constater qu'il n'existe pas de stratégie qui soit meilleure que les autres pour *toutes* les associations modèle/jeu de données. Partant de ce constat, plusieurs benchmarks réalisés sur de nombreux jeux de données ont mis en évidence les stratégies qui, en moyenne, sont les plus adaptées pour les modèles de classification supervisée de l'état de l'art (Yang et Loog, 2018; Pereira-Santos et al., 2019).

Ce constat a également motivé des travaux récents visant à définir des méthodes de méta-apprentissage permettant d'*apprendre* de nouvelles stratégies d'apprentissage actif à partir de jeux de données déjà étiquetés (Konyushkova et al., 2019). On s'attend alors à ce qu'une stratégie apprise soit en mesure de contrôler automatiquement le compromis exploitation/exploration lors de son utilisation sur des jeux de données non étiquetés nouveaux, ce que ne permettent pas les stratégies heuristiques de l'état de l'art.

L'objectif de cet article est de comparer deux options possibles dans la réalisation d'une solution d'apprentissage actif utilisable en contexte industriel : utiliser une stratégie heuristique qui, en moyenne, est la meilleure pour un modèle donné, ou bien exploiter une stratégie apprise selon une approche *data-driven*. Pour cela, nous présentons les résultats d'un benchmark réalisé sur 20 jeux de données qui compare une stratégie apprise selon la méthode proposée dans (Konyushkova et al., 2019) avec un échantillonnage basé sur une mesure d'incertitude, les modèles utilisés étant dans les deux cas la régression logistique et la forêt aléatoire.

2 Apprentissage de stratégies d'apprentissage actif

Dans cette section, nous explicitons l'ensemble des aspects de la méthode LAL proposée dans (Konyushkova et al., 2019), à savoir la modélisation de l'apprentissage actif comme processus de décision markovien et l'utilisation du Deep Q-Learning pour l'apprentissage de la stratégie.

2.1 L'apprentissage actif comme processus de décision markovien

La méthode LAL repose sur la formulation de l'apprentissage actif comme processus de décision markovien, qui est assez naturelle. Dans chaque *état* du processus, l'*agent* effectue une *action*, qui est la sélection d'une instance à étiqueter, et celui-ci reçoit une *récompense* qui dépend de la qualité du modèle appris avec la nouvelle instance. La stratégie d'apprentissage actif devient la *politique* suivie par l'agent qui associe une action à un état.

Dans ce cadre, l'itération t du processus d'apprentissage de la politique à partir d'un jeu de données divisé en un ensemble d'apprentissage $\mathcal{D} = \mathcal{L}_t \cup \mathcal{U}_t$ et un ensemble de test \mathcal{D}' est composée des étapes suivantes :

1. Un modèle $h^{(t)}$ est appris à partir de \mathcal{L}_t . Associé à \mathcal{L}_t et \mathcal{U}_t , il permet de caractériser un état s_t .
2. L'agent effectue l'action $a_t = \pi(s_t) \in \mathcal{A}_t$ qui définit l'instance $x^{(t)} \in \mathcal{U}_t$ à étiqueter.

3. L'étiquette $y^{(t)}$ de $\mathbf{x}^{(t)}$ est récupérée, puis l'ensemble d'apprentissage est mis à jour, *i.e.* $\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{(\mathbf{x}^{(t)}, y^{(t)})\}$ et $\mathcal{U}_{t+1} = \mathcal{U}_t \setminus \{\mathbf{x}^{(t)}\}$.
4. L'agent reçoit la récompense r_t associée à la performance ℓ_t sur l'ensemble de test \mathcal{D}' . Celle-ci est exploitée pour mettre à jour la politique (voir Section 2.2).

On notera que l'ensemble des actions \mathcal{A}_t dépend ici du temps car il n'est pas possible de sélectionner plusieurs fois une même instance. Ces étapes sont répétées jusqu'à ce qu'un état terminal s_T soit atteint. Ici, on considère que l'on est dans un état terminal lorsque toutes les instances ont été étiquetées ou lorsque $\ell_t \geq q$, où q est un seuil de performance qui a été choisi comme étant 98% de la performance obtenue lorsque le modèle est appris sur toutes les données d'apprentissage.

La définition précise de l'ensemble des états, de l'ensemble des actions et de la fonction de récompense n'est pas évidente. Pour définir un état, il a été proposé d'utiliser un vecteur dont les composantes sont les scores $\hat{y}_t(\mathbf{x}) = \mathbb{P}(Y = 0 | \mathbf{x})$ associés aux instances non étiquetées d'un sous-ensemble \mathcal{V} mis de côté. Il s'agit de la représentation la plus simple permettant de caractériser l'incertitude d'un classifieur sur un jeu de données à un instant donné t .

L'ensemble des actions a été défini à l'itération t comme l'ensemble des vecteurs $\mathbf{a}_i = [\hat{y}_t(\mathbf{x}_i), g(\mathbf{x}_i, \mathcal{L}_t), g(\mathbf{x}_i, \mathcal{U}_t)]$, où $\mathbf{x}_i \in \mathcal{U}_t$ et :

$$g(\mathbf{x}_i, \mathcal{L}_t) = \frac{1}{|\mathcal{L}_t|} \sum_{\mathbf{x}_j \in \mathcal{L}_t} \text{dist}(\mathbf{x}_i, \mathbf{x}_j), \quad g(\mathbf{x}_i, \mathcal{U}_t) = \frac{1}{|\mathcal{U}_t|} \sum_{\mathbf{x}_j \in \mathcal{U}_t} \text{dist}(\mathbf{x}_i, \mathbf{x}_j),$$

où dist est la distance du cosinus. Une action est donc caractérisée par l'incertitude du classifieur sur l'instance associée, ainsi que par deux statistiques liées à la densité du voisinage de l'instance.

La fonction de récompense a été choisie constante et négative jusqu'à l'arrivée dans un état terminal ($r_t = -1$). Ainsi, pour maximiser sa récompense, l'agent doit effectuer le moins d'interactions possible.

2.2 Apprentissage de la stratégie par Deep Q-Learning

La méthode LAL utilise l'algorithme du Deep Q-Learning pour l'apprentissage de la politique optimale. Pour pouvoir traiter un espace d'états qui évolue à chaque itération, l'architecture du réseau neuronal a été adaptée. La nouvelle architecture considère les actions comme des entrées de la fonction Q au même titre que les états. Elle ne renvoie alors qu'une seule valeur, là où l'architecture classique ne prend en entrée qu'un état et renvoie les valeurs associées à toutes les actions.

L'apprentissage est effectué sur une collection de Z jeux de données étiquetées $\{\mathcal{Z}_i\}_{1 \leq i \leq Z}$. Il consiste à répéter la procédure suivante (voir Figure 1) :

1. Un jeu de données $\mathcal{Z} \in \{\mathcal{Z}_i\}$ est sélectionné aléatoirement et divisé en un ensemble d'apprentissage \mathcal{D} et un ensemble de test \mathcal{D}' .
2. La politique π dérivée du Deep Q-Network est utilisée pour simuler plusieurs épisodes d'apprentissage actif sur \mathcal{Z} selon la procédure décrite à la Section 2.1. Les expériences $(\mathbf{s}_t, \mathbf{a}_t, r_t, \mathbf{s}_{t+1})$ sont collectées dans une mémoire de taille finie.
3. Les paramètres du Deep Q-Network sont mis à jour plusieurs fois à partir d'un *mini-batch* d'expériences extrait de la mémoire.

Apprentissage par renforcement de stratégies d'apprentissage actif : une évaluation

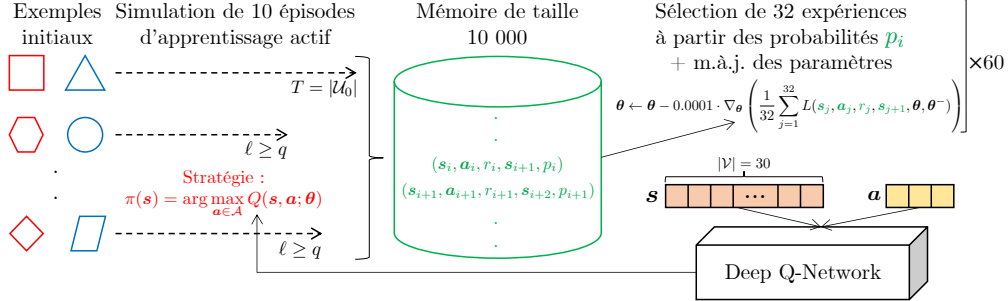


FIG. 1 – Représentation des différentes étapes intervenant dans une itération de la phase d'apprentissage de la politique par Deep Q-Learning.

Pour initialiser le Deep Q-Network, des épisodes de *warm start* sont simulés avec une politique d'échantillonnage aléatoire, suivis de plusieurs mises à jour des paramètres.

3 Protocole de l'étude expérimentale

Dans cette section, nous explicitons les différents aspects du protocole de l'étude expérimentale comparative que nous avons réalisée.

Apprentissage de la stratégie. Pour l'apprentissage de la stratégie, nous avons repris le même code¹, les mêmes hyper-paramètres et les mêmes 8 jeux de données que ceux utilisés dans (Konyushkova et al., 2019).

Associations heuristique/modèle évaluées. Notre objectif est de comparer les performances d'une stratégie apprise *via* la méthode LAL avec celles d'une stratégie heuristique qui, en moyenne, donne les meilleurs résultats pour un modèle donné. Plusieurs benchmarks réalisés sur de nombreux jeux de données ont mis en évidence le fait que l'échantillonnage basé sur l'incertitude est la meilleure heuristique pour la régression logistique (LR) et la forêt aléatoire (RF) (Yang et Loog, 2018; Pereira-Santos et al., 2019). Cette heuristique consiste à choisir l'instance pour laquelle la différence (ou marge) entre les probabilités des deux classes les plus probables est la plus petite. Bien que performante, son défaut majeur est son absence totale d'exploration.

Nous avons choisi d'évaluer l'association Margin/LR car c'est avec la régression logistique que les hyper-paramètres du Deep Q-Learning ont été optimisés dans (Konyushkova et al., 2019). De plus, dans l'objectif de déterminer s'il est nécessaire de modifier ces derniers lorsque l'on change de modèle, nous avons aussi évalué l'association Margin/RF. Cette dernière association est d'ailleurs particulièrement intéressante dans la mesure où il s'agit de la meilleure association mise en évidence dans un benchmark réalisé sur 73 jeux de données, incluant 5 modèles de classification et 8 stratégies d'apprentissage actif (Pereira-Santos et al., 2019). En outre, nous avons évalué l'échantillonnage aléatoire (Rnd) pour les deux modèles.

1. <https://github.com/ksenia-konyushkova/LAL-RL>

Jeux de données. Les jeux de données retenus pour l'étude ont été sélectionnés de sorte à avoir une grande diversité selon les critères suivants : (i) nombre d'individus ; (ii) nombre de variables quantitatives ; (iii) nombre de variables qualitatives ; (iv) déséquilibre entre les classes. Nous avons également veillé à exclure les jeux de données trop petits, qui ne sont pas représentatifs de ceux qui sont exploités en contexte industriel. Les 20 jeux de données sélectionnés sont décrits dans le Tableau 1. Ceux-ci sont tous issus de la base UCI (Dua et Graff, 2017), à l'exception du jeu de données « orange-fraud ».

Jeux de données	$ \mathcal{D} $	$ \mathcal{Y} $	#num	#nom	maj (%)	min (%)
adult	48790	2	6	8	76.06	23.94
banana	5292	2	2	0	55.16	44.84
bank-marketing-full	45211	2	7	9	88.30	11.70
climate-simulation-craches	540	2	20	0	91.48	8.52
eeg-eye-state	14980	2	14	0	55.12	44.88
hiva	40764	2	1617	0	96.50	3.50
ibn-sina	13951	2	92	0	76.18	23.82
magic	18905	2	10	0	65.23	34.77
musk	6581	2	166	1	84.55	15.45
nomao	32062	2	89	29	69.40	30.60
orange-fraud	1680	2	16	0	63.75	36.25
ozone-onehr	2528	2	72	0	97.11	2.89
qsar-biodegradation	1052	2	41	0	66.35	33.65
seismic-bumps	2578	2	14	4	93.41	6.59
skin-segmentation	51444	2	3	0	71.51	28.49
statlog-german-credit	1000	2	7	13	70.00	30.00
thoracic-surgery	470	2	3	13	85.11	14.89
thyroid-hypothyroid	3086	2	7	18	95.43	4.57
wilt	4819	2	5	0	94.67	5.33
zebra	61488	2	154	0	95.42	4.58

TAB. 1 – *Jeux de données retenus pour l'évaluation de la stratégie apprise par la méthode LAL. Colonnes : nombre d'exemples, nombre de classes, nombres de variables quantitatives et qualitatives, proportions d'exemples dans les classes majoritaire et minoritaire.*

Critère d'évaluation. Dans notre protocole d'évaluation, le processus d'échantillonnage actif débute par la sélection aléatoire d'une instance dans chaque classe et s'achève lorsque 250 instances ont été étiquetées. Cette valeur a été choisie pour nous assurer d'avoir des résultats comparables aux autres études comparatives de la littérature. L'*accuracy* et l'aire sous la courbe d'apprentissage (ALC²) ont respectivement été retenus comme mesures de la performance d'un classifieur et d'une stratégie. L'ALC basée sur l'*accuracy* est la mesure la plus utilisée en apprentissage actif, et elle a notamment été choisie dans le cadre d'un *challenge* (Guyon et al., 2011).

L'évaluation a été effectuée par validation croisée avec 5 partitions, au sein desquelles le déséquilibre entre les classes au sein du jeu de données complet a été préservé. Pour chaque partition, le processus d'échantillonnage a été répété 5 fois avec des initialisations différentes

2. [Area under the Learning Curve]

pour obtenir une moyenne et une variance sur le résultat. Nous avons toutefois veillé à ce que les instances initiales soient identiques pour toutes les associations stratégie/modèle sur chaque partition pour ne pas introduire de biais dans les résultats. De plus, pour la stratégie Rnd, la suite aléatoire de nombres a été identique pour tous les modèles.

4 Résultats

Les résultats de nos expérimentations sont donnés dans le Tableau 2. Les ALC moyennes obtenues pour chaque association jeu de données/classifieur/stratégie y sont reportées (la valeur maximale est 100). La partie gauche du tableau donne les résultats pour une régression logistique et la partie droite donne les résultats pour une forêt aléatoire. L'avant-dernière ligne correspond aux moyennes calculées sur tous les jeux de données et la dernière ligne donne le nombre de fois où la stratégie est arrivée gagnante (*win*), à égalité (*tie*) ou perdante (*loss*). Les différences non significatives ont été établies *via* un test de Student apparié avec $\alpha = 0.01$.

Jeu de données	Rnd/LR	Margin/LR	LAL/LR	Rnd/RF	Margin/RF	LAL/RF	maj (%)
adult	77.93	78.91	78.97	80.17	81.27	81.21	76.06
banana	53.03	57.39	53.12	80.24	73.81	73.58	55.16
bank-marketing-full	86.85	87.62	87.72	88.19	88.34	88.49	88.30
climate-simulation-craches	87.22	89.13	88.62	91.15	91.14	91.13	91.48
eeg-eye-state	56.08	55.32	56.11	65.53	67.58	64.42	55.12
hiva	64.43	70.84	71.80	96.32	96.47	96.44	96.50
ibn-sina	84.77	88.58	88.90	90.53	93.41	92.75	76.18
magic	76.49	77.93	77.64	78.05	80.79	79.68	65.23
musk	83.73	82.34	81.95	89.55	96.18	95.35	84.55
nomao	89.45	91.43	91.37	89.41	92.32	92.07	69.40
orange-fraud	76.70	81.74	74.26	89.15	90.66	90.48	63.75
ozone-onehr	92.90	94.26	95.06	96.61	96.83	96.89	97.11
qsar-biodegradation	80.98	82.62	83.53	80.34	82.76	82.40	66.35
seismic-bumps	90.87	92.59	92.14	92.48	92.92	93.02	93.41
skin-segmentation	77.05	82.69	83.21	91.51	95.70	95.77	71.51
statlog-german-credit	70.76	72.12	72.34	72.25	72.93	72.78	70.00
thoracic-surgery	83.76	83.93	82.72	83.51	84.41	84.18	85.11
thyroid-hypothyroid	97.21	97.99	97.97	97.75	98.77	98.71	95.43
wilt	93.53	95.18	92.87	94.86	97.23	97.02	94.67
zebra	86.40	90.31	91.36	94.71	95.54	95.25	95.42
Moyenne	80.51	82.65	82.08	87.12	88.45	88.08	79.53
win/tie/loss	0/5/15	3/15/2	2/15/3	1/4/15	3/16/1	0/16/4	

TAB. 2 – Résultats de nos expérimentations.

Plusieurs observations peuvent être formulées. Tout d'abord, il faut noter que le choix du modèle est déterminant : les résultats de la forêt aléatoire sont tous meilleurs que ceux de la régression logistique. La forêt aléatoire est en effet un modèle qui apprend bien à partir de peu de données, ce qui avait déjà été souligné dans (Salperwyck et Lemaire, 2011). Adjoindre à ce modèle la stratégie Margin permet de plus une amélioration significative des performances.

En outre, l'examen des résultats de la stratégie apprise *via* la méthode LAL montre très clairement qu'une bonne stratégie d'apprentissage actif a été apprise, celle-ci obtenant de meilleures performances que l'échantillonnage aléatoire sur un grand nombre de jeux de don-

nées. La stratégie apprise n’est cependant pas meilleure que l’échantillonnage basé sur l’incertitude. Ces résultats sont néanmoins très intéressants dans la mesure où seuls 8 jeux de données ont été utilisés pour l’apprentissage de la stratégie.

Les résultats exhibent de plus un fait connu concernant l’apprentissage actif : sur des jeux de données très déséquilibrés, il est difficile d’atteindre une performance meilleure que celle de la stratégie Rnd, ce que montre la dernière colonne du Tableau 2 dans laquelle se trouvent les résultats obtenus en prédisant toujours la classe majoritaire. Toutefois, si le déséquilibre est connu, il pourrait être intéressant d’associer aux stratégies un modèle ou un critère approprié à ce cas, comme illustré dans (Ertekin et al., 2007).

Enfin, l’analyse des résultats à plusieurs instants du processus d’échantillonnage ne montre pas de variabilité dans nos conclusions (voir Tableau 3). Nos résultats pour $|\mathcal{L}| = 32$ confirment par ailleurs les résultats de (Salperwyck et Lemaire, 2011). Tous nos résultats expérimentaux sont disponibles sur Github³.

$ \mathcal{L} = 32$			$ \mathcal{L} = 64$			$ \mathcal{L} = 128$			$ \mathcal{L} = 250$		
Rnd	Margin	LAL	Rnd	Margin	LAL	Rnd	Margin	LAL	Rnd	Margin	LAL
84.60	85.84	85.42	86.51	88.07	87.58	88.08	89.56	89.17	89.42	90.55	90.40

TAB. 3 – Accuracy moyenne pour différentes valeurs de $|\mathcal{L}|$ (forêt aléatoire).

5 Conclusion

Dans cet article, nous avons évalué une méthode représentative d’une orientation récente de la recherche en apprentissage actif vers des méthodes de méta-apprentissage visant à « apprendre à apprendre activement ». Plusieurs enseignements peuvent être tirés de notre étude.

Tout d’abord, les expérimentations réalisées confirment l’intérêt de la méthode LAL, dans la mesure où elle a permis d’apprendre une stratégie atteignant les performances d’une très bonne heuristique, à savoir un échantillonnage basé sur l’incertitude. Il faut cependant noter que cette méthode soulève encore de nombreux problèmes, dont celui de la transférabilité des stratégies apprises. Une solution d’apprentissage actif utilisable en contexte industriel doit en effet être performante sur des données réelles dont la nature est inconnue et ne doit pas impliquer de paramètres à ajuster. S’agissant de la méthode LAL, un premier problème majeur est donc celui de la constitution d’un *dataset* de *datasets* suffisamment grand et varié pour que la stratégie apprise soit performante dans des contextes très différents.

De plus, la procédure d’apprentissage de la stratégie est sensible au critère de performance utilisé, ce qui nous semble poser problème. Il faudrait idéalement que la stratégie apprise soit utilisable sur de nouveaux jeux de données avec un critère de performance arbitraire (AUC, F-score, etc.). Il n’est pas envisageable, de notre point de vue, que le travail d’optimisation des nombreux hyper-paramètres de la méthode soit effectué par un utilisateur n’ayant pas d’expertise en apprentissage par renforcement profond.

Outre l’évaluation de la méthode LAL, notre étude a permis de confirmer un résultat de (Pereira-Santos et al., 2019), à savoir que l’échantillonnage basé sur l’incertitude, associé à

3. https://github.com/l-desreumaux/lal_evaluation

une forêt aléatoire, est une stratégie très compétitive. Pour autant, celle-ci partage avec toutes les stratégies d'apprentissage actif un défaut majeur, qui est son absence de fiabilité. En effet, il n'existe pas de stratégie qui soit meilleure ou équivalente à l'échantillonnage aléatoire sur tous les jeux de données et avec tous les modèles. La littérature en apprentissage actif est lacunaire s'agissant de cette problématique, qui est pourtant un verrou majeur à une utilisation en situation réelle. Un autre problème important, peu étudié dans la littérature, est celui de l'estimation de l'erreur de généralisation en l'absence d'un ensemble de test.

Les benchmarks récents ont mis en évidence la nécessité d'avoir recours à des expérimentations exhaustives pour comparer des stratégies d'apprentissage actif. La communauté de chercheurs gagnerait peut-être à se munir d'un benchmark de référence, comme dans le domaine de la classification de séries temporelles (Chen et al., 2015), de sorte à ce que les nouveaux résultats puissent être rigoureusement comparés à l'état de l'art.

Références

- Chen, Y., E. Keogh, B. Hu, N. Begum, A. Bagnall, A. Mueen, et G. Batista (2015). The UCR Time Series Classification Archive. www.cs.ucr.edu/~eamonn/time_series_data/.
- Dua, D. et C. Graff (2017). UCI machine learning repository.
- Ertekin, S., J. Huang, L. Bottou, et L. Giles (2007). Learning on the Border : Active Learning in Imbalanced Data Classification. In *Conference on Information and Knowledge Management*, CIKM, pp. 127–136.
- Guyon, I., G. Cawley, G. Dror, et V. Lemaire (2011). Results of the Active Learning Challenge. In *Proceedings of Machine Learning Research*, Volume 16, pp. 19–45. PMLR.
- Konyushkova, K., R. Sznitman, et P. Fua (2019). Discovering General-Purpose Active Learning Strategies. *arXiv:1810.04114 [cs.LG]*.
- Pereira-Santos, D., R. B. C. Prudêncio, et A. C. de Carvalho (2019). Empirical investigation of active learning strategies. *Neurocomputing* 326–327, 15–27.
- Salperwyck, C. et V. Lemaire (2011). Learning with few examples : An empirical study on leading classifiers. In *International Joint Conference on Neural Networks*, pp. 1010–1019.
- Settles, B. (2012). *Active Learning*. Morgan & Claypool Publishers.
- Yang, Y. et M. Loog (2018). A benchmark and comparison of active learning for logistic regression. *Pattern Recognition* 83, 401–415.

Summary

Active learning aims to reduce annotation cost by predicting which samples are useful for a human expert to label. In the literature, most selection strategies are hand-designed, and it has become clear that there is no best active learning strategy. This has motivated research into meta-learning algorithms for “learning how to actively learn” (Konyushkova et al., 2019). In this paper, we compare this approach with margin sampling, reported in recent comparative studies as a very competitive heuristic (Yang et Loog, 2018; Pereira-Santos et al., 2019).