Apprentissage contrastif pour l'adaptation de domaine en régression

Mohamad DHAINI[1,2], Maxime BERAR[1], Paul HONEINE[1], Antonin VAN EXEM[2]

[1] Univ Rouen Normandie, INSA Rouen Normandie, Université Le Havre Normandie, Normandie Univ, LITIS UR 4108, F-76000 Rouen, France

[2]Tellux, 76650 Petit-Couronne, France

mohamad.dhaini@univ-rouen.fr

Résumé – L'adaptation de domaine non supervisée relève le défi d'utiliser des modèles d'apprentissage statistique sur des données de distribution différente de celle des données d'entraînement. Cela impose d'apprendre des représentations efficaces qui peuvent être généralisées à travers les domaines. Dans cet article, nous étudions l'utilisation de l'apprentissage contrastif pour améliorer les approches d'adaptation de domaine. À cette fin, l'apprentissage contrastif est appliqué à l'espace latent d'un réseau de neurones, où l'objectif est d'apprendre une représentation qui maximise la similitude entre des exemples similaires et minimise la similitude entre des exemples dissemblables. En outre, pour minimiser l'écart entre les domaines source et cible, le procédé utilise l'apprentissage par dictionnaire, où les dictionnaires sont extraits à la fois des données source et cible et la trajectoire entre les deux dictionnaires est minimisée. La méthode proposée est évaluée sur le jeu de données *dSprites*, montrant de meilleures performances que les méthodes de l'état de l'art.

Abstract – Unsupervised domain adaptation addresses the challenge of using Machine Learning models on data from a different distribution than that of the training data. This imposes learning effective representations that can generalize across domains. In this paper, we investigate contrastive learning to improve domain adaptation. For this purpose, contrastive learning is applied to the latent space of a neural network, where the goal is to learn a representation that maximizes the similarity between similar examples and minimizes the similarity between dissimilar ones. Furthermore, to minimize the gap between the source and target domains, the method utilizes dictionary learning, where dictionaries are extracted from both source and target data and the trajectory between both dictionaries is minimized. The proposed method is evaluated on the *dSprites* dataset, where the results show that it outperforms existing unsupervised domain adaptation methods.

1 Introduction

En apprentissage statistique, l'adaptation de domaine est une tâche cruciale, dès lors que l'objectif est d'entraîner des architectures de réseaux de neurones pour fonctionner de manière aussi performante sur un domaine cible, que sur un domaine source différent où sont disponibles les données étiquetées. Dans ce cadre totalement non supervisé, où aucune donnée étiquetée n'est disponible dans le domaine cible, les approches basées sur les caractéristiques visent à apprendre une représentation latente jointe aux deux domaines et qui sera ensuite transmise dans la partie tâche du réseau. Parmi ces méthodes, celles de sous-espace cherchent à aligner les sous-espaces principaux de chaque domaine par décomposition en valeurs singulières (SVD) [1] ou par apprentissage de dictionnaires (DL) [2]. D'autres méthodes reposent sur l'alignement des distributions des données source et cible [3]. Cependant leur efficacité dépend du nombre de données disponibles et l'on peut chercher à augmenter l'ensemble d'entraînement.

Une manière d'augmenter le nombre de données est d'appliquer des transformations limitées aux entrées (par exemple tourner des images). Comme ces variations s'appuient sur une même donnée d'entrée, il est intéressant d'utiliser l'apprentissage contrastif. En maximisant la similitude entre des exemples similaires et en minimisant celle entre des exemples dissemblables, l'apprentissage contrastif peut apprendre la structure sous-jacente des données au delà des transformations imposées. Plusieurs travaux ont utilisé l'apprentissage contrastif pour améliorer l'adaptation de domaine non supervisée, comme [4] où une métrique modélisant explicitement la divergence entre le domaine intra-classe et la divergence de domaine inter-classes est utilisée. En outre, [5] propose une nouvelle fonction de perte contrastive pour la régression dont le but est de rapprocher les exemples similaires et d'éloigner ceux différents, en reposant sur une classification monoclasse.

Le présent article vise à améliorer ces propositions, en proposant deux versions de pertes contrastives. La première, basée sur l'entropie croisée, adapte la fonction de perte contrastive conventionnelle, utilisée en classification, pour les problèmes de régression. Cette adaptation repose sur un seuil de similarité qui permet l'étiquetage des paires positives et négatives. Ce seuil est uniquement applicable au domaine source. Pour la deuxième fonction de perte, nous proposons une relaxation pour ne plus avoir besoin de seuil pour étiqueter les exemples et donc applicable aux deux domaines. Dans le cadre de la mise en œuvre globale de l'apprentissage, la fonction coût liée à l'adaptation de domaine que nous utiliserons est basée sur l'apprentissage du dictionnaire [2].

État de l'art

Au delà de l'adaptation de domaine dans un espace de Hilbert à noyau reproduisant (RKHS) par la divergence MMD [6], les récentes avancées reposent sur les réseaux de neurones profonds. Dans [3], la fonction de prédiction est adaptée, du domaine source au domaine cible, sous l'hypothèse de l'existence d'une transformation non linéaire entre les distributions conjointes de caractéristiques et d'étiquettes des deux domaines. La méthode *deep adaptation network* (DAN) [7] intègre les représentations latentes dans un RKHS, explicitant la correspondance entre les moyennes des distributions des 2 domaines ; l'écart entre les domaines est réduit à l'aide d'une méthode de sélection multi-noyaux optimale. De la même manière, la méthode *domain-adversarial neural networks* (DANN) [8] utilise une perte antagoniste pour apprendre une représentation invariante entre les domaines. Dans la méthode *maximum classifier discrepancy* (MCD) [9], des caractéristiques cibles sont générées près de la frontière de décision afin de minimiser l'écart de domaine tout en maximisant l'écart entre les sorties des deux classificateurs.

L'adaptation de domaine en régression est plus difficile qu'en classification, car elle nécessite des caractéristiques robustes au changement d'échelle [1, 2]. Pour surmonter cette difficulté, la méthode *representation subspace distance* (RSD) utilise une base de l'espace latent extraite par SVD et alignée entre les deux domaines [1]. Dans [2], nous avons proposé une nouvelle méthode d'adaptation de domaine non supervisée basée sur l'apprentissage du dictionnaire, qui est compatible avec la rétropropagation et peut être mise en œuvre dans des réseaux profonds de bout en bout. Dans cet article, nous revisitons ce formalisme à la lumière de l'apprentissage contrastif.

2 Méthode proposée

Dans cette section, nous décrivons la méthode d'adaptation de domaine via l'apprentissage conjoint du dictionnaire et de représentation par minimisation d'une perte contrastive.

2.1 Adaptation de domaine par apprentissage du dictionnaire

Soit $\mathbb{R}^k$ l'espace d'entrée. Le réseau prend en entrée un lot de $N_{\mathcal{S}}$ échantillons de l'ensemble de données source et $N_{\mathcal{T}}$ de l'ensemble de données cible, désignés respectivement par :

$$\begin{cases} \mathbf{X}_{\mathcal{S}} = [x_{\mathcal{S}}^1, x_{\mathcal{S}}^2, \ldots, x_{\mathcal{S}}^{N_{\mathcal{S}}}]^\top \in \mathbb{R}^{N_{\mathcal{S}} \times k}, \\ \mathbf{X}_{\mathcal{T}} = [x_{\mathcal{T}}^1, x_{\mathcal{T}}^2, \ldots, x_{\mathcal{T}}^{N_{\mathcal{T}}}]^\top \in \mathbb{R}^{N_{\mathcal{T}} \times k}. \end{cases}$$

Soit $y_{\mathcal{S}}^i$ l'étiquette du i-ième échantillon $x_{\mathcal{S}}^i$ du lot source, avec $\mathbf{Y}_{\mathcal{S}} = [y_{\mathcal{S}}^1, y_{\mathcal{S}}^2, \ldots, y_{\mathcal{S}}^{N_{\mathcal{S}}}]^\top$. La première partie du réseau de neurones consiste en un encodeur Φ_w, de paramètres w, qui lie l'espace d'entrée $\mathbb{R}^k$ vers un espace latent $\mathbb{R}^b$. Les caractéristiques extraites du lot source $\mathbf{X}_{\mathcal{S}}$ et du lot cible $\mathbf{X}_{\mathcal{T}}$ sont notées respectivement

$$\mathbf{F}_{\mathcal{S}} = \Phi_w(\mathbf{X}_{\mathcal{S}}) \in \mathbb{R}^{N_{\mathcal{S}} \times b} \text{ et } \mathbf{F}_{\mathcal{T}} = \Phi_w(\mathbf{X}_{\mathcal{T}}) \in \mathbb{R}^{N_{\mathcal{T}} \times b}.$$

Soient $f_{\mathcal{S}}^i$ et $f_{\mathcal{T}}^i$ les deux caractéristiques obtenues à partir des échantillons $x_{\mathcal{S}}^i$ et $x_{\mathcal{T}}^i$, respectivement. La première étape consiste à apprendre un dictionnaire (DL) du domaine source par $\mathbf{F}_{\mathcal{S}}$. Le dictionnaire $\mathbf{D}_{\mathcal{S}} = [d_1, ..., d_m]^\top$ de m atomes et son codage $\mathbf{R}_{\mathcal{S}}$ sont donnés par le problème d'optimisation

$$\begin{cases} (\mathbf{D}_{\mathcal{S}}, \mathbf{R}_{\mathcal{S}}) = \underset{\mathbf{D} \in \mathcal{C}, \mathbf{R} \in \mathbb{R}^{N_{\mathcal{S}} \times m}}{\operatorname{argmin}} \|\mathbf{F}_{\mathcal{S}} - \mathbf{R}\mathbf{D}\|_F^2 + \lambda \|\mathbf{R}\|_1 \\ \text{où } \mathcal{C} = \left\{\mathbf{D} \in \mathbb{R}^{m \times b} \ : \|d_i\|_2 \leq 1, \forall i = 1, \ldots, m\right\} \end{cases} \tag{1}$$

où λ contrôle le niveau de parcimonie. Une stratégie bien connue pour résoudre ce problème consiste à utiliser un algorithme de descente de gradient projeté [2]. Ayant le dictionnaire source $\mathbf{D}_{\mathcal{S}}$, on cherche alors à reconstruire les entités cibles $\mathbf{F}_{\mathcal{T}}$ avec $\mathbf{D}_{\mathcal{S}}$ via une décomposition parcimonieuse comme suit :

$$\mathbf{R}_{\mathcal{T}} = \underset{\mathbf{R} \in \mathbb{R}^{N_{\mathcal{T}} \times m}}{\arg\min} \|\mathbf{F}_{\mathcal{T}} - \mathbf{R}\mathbf{D}_{\mathcal{S}}\|_F^2 + \gamma \|\mathbf{R}\|_{\mathbf{1}} \tag{2}$$

où γ est le niveau de parcimonie. La résolution de ce problème d'optimisation se fait à l'aide de l'algorithme FISTA. Le résidu de cette reconstruction peut être exprimé comme suit :

$$\mathbf{J}_{res} = \mathbf{F}_{\mathcal{T}} - \mathbf{R}_{\mathcal{T}}\mathbf{D}_{\mathcal{S}} \tag{3}$$

Afin que $\mathbf{D}_{\mathcal{S}}$ décrive mieux le domaine cible, on vise à réduire la norme du résidu $\mathbf{J}_{res}$. L'ajustement optimal $\Delta\mathbf{D}_{\mathcal{S}}$ est

$$\Delta\mathbf{D}_{\mathcal{S}} = \left(\mathbf{R}_{\mathcal{T}}^\top \mathbf{R}_{\mathcal{T}} + \alpha\mathbf{I}\right)^{-1} \mathbf{R}_{\mathcal{T}}^\top \mathbf{J}_{res}, \tag{4}$$

où $\mathbf{I}$ est la matrice d'identité et α est un paramètre de compromis. Par conséquent, on définit la perte de domaine selon :

$$\mathcal{L}_{\text{Dom}} = \|\Delta\mathbf{D}_{\mathcal{S}}\|_F^2 \tag{5}$$

2.2 Apprentissage contrastif

En apprentissage contrastif, l'objectif est de contraindre un réseau de neurones à trouver un espace de représentation où les données similaires (similaires au sens de la sortie désirée) sont proches les unes des autres et celles dissemblables éloignées. Cette similarité entre points est induite par le biais de l'augmentation de données, ainsi la première étape consiste à créer différentes versions d'un échantillon. Cela peut être fait en utilisant diverses techniques telles que le recadrage aléatoire, les rotations aléatoires ou la gigue de couleurs. En appliquant la transformation aux données source et cible, nous obtenons

$$\widetilde{\mathbf{X}}_{\mathcal{S}} = \Phi_{transform}(\mathbf{X}_{\mathcal{S}}) \text{ et } \widetilde{\mathbf{X}}_{\mathcal{T}} = \Phi_{transform}(\mathbf{X}_{\mathcal{T}}).$$

En passant $\widetilde{\mathbf{X}}_{\mathcal{S}}$ et $\mathbf{X}_{\mathcal{S}}$ à Φ_w, nous obtenons les caractéristiques extraites $\widetilde{\mathbf{F}}_{\mathcal{S}}$ et $\mathbf{F}_{\mathcal{S}}$ respectivement. L'objectif de la perte contrastive est de rassembler les paires positives et de repousser les paires négatives disponibles dans le vecteur concaténé de $\widetilde{\mathbf{F}}_{\mathcal{S}}$ et $\mathbf{F}_{\mathcal{S}} \in \mathbb{R}^{2N_{\mathcal{S}} \times b}$. En classification, la sélection de paires positives est effectuée en prenant la version transformée d'une image ainsi que d'autres images appartenant à la même classe, tout en traitant le reste comme des paires négatives. En régression, l'alternative est de définir pour le i-ème échantillon une boule $\mathbf{B}^i$ de rayon $\mathbf{r}$ où la paire positive j est sélectionnée comme suit :

$$\mathbf{r} \geq \left\|y^i - y^j\right\|_2 \tag{6}$$

La perte contrastive souvent utilisée est basée sur l'entropie croisée qui, pour notre problème de régression, devient

$$\mathcal{L}^{\mathcal{S}}_{\text{EContrast}} = -\frac{1}{N}\sum_{i=1}^{2N}\sum_{j\in\mathbf{B}^i}\log\frac{\exp\left(\text{sim}\left(f^i,f^j\right)/\tau\right)}{\sum_{k\notin\mathbf{B}^i}\exp\left(\text{sim}\left(f^i,f^k\right)/\tau\right)}, \tag{7}$$

où $\text{sim}(\boldsymbol{u},\boldsymbol{v}) = \boldsymbol{u}^T\boldsymbol{v}/(\|\boldsymbol{u}\|\|\boldsymbol{v}\|)$ est la similarité cosinus entre deux vecteurs et $\tau \in \mathbb{R}_+$ la température. La similarité cosinus est une mesure angulaire entre les vecteurs normalisés. Cependant, dans les problèmes de régression, la fonction de prédiction est très sensible à la norme des caractéristiques et au changement d'échelle [1, 2].Pour atteindre cet objectif, nous utilisons une perte contrastive alternative décrite dans [10],

$$\min_{\Phi_w}\sum_{i=1}^{2N}\Big(\sum_{j\in\mathbf{B}^i}\text{D}\left(f^i,f^j\right)-\lambda\sum_{k\notin\mathbf{B}^i}\text{D}\left(f^i,f^k\right)\Big), \tag{8}$$

où D est une distance. La distance euclidienne peut être adéquate en cas d'apprentissage contrastif en raison des gradients forts et lisses qu'elle présente par rapport aux autres distances. Le deuxième terme de la perte contrastive dans (8) tente de maximiser la distance entre les caractéristiques des paires négatives. Cependant, ce terme est redondant avec la perte de régression définie dans le problème. Nous proposons donc une nouvelle version de perte contrastive ne retenant que la minimisation des distances entre paires positives comme suit :

$$\min_{\Phi_w}\sum_{i=1}^{2N}\sum_{j\in\mathbf{B}^i}\text{D}\left(f^i,f^j\right) \tag{9}$$

Ces pertes reposent sur un paramètre de rayon $\mathbf{r}$ pour définir la boule $\mathbf{B}$. Cela rend le comportement de la perte contrastive fortement dépendant du choix de $\mathbf{r}$. Pour surmonter ce problème, nous considérons la version transformée $\widetilde{x}^i$ de x^i comme sa seule paire positive et minimisons la distance entre $\widetilde{f}^i$ et f^j. En conséquence, une version simplifiée de la fonction de perte contrastive peut être décrite comme la norme de Frobenius entre les matrices de caractéristiques d'origine et transformées comme suit :

$$\mathcal{L}^{\mathcal{S}}_{\text{SContrast}} = \left\|\widetilde{\mathbf{F}}_{\mathcal{S}} - \mathbf{F}_{\mathcal{S}}\right\|_F \tag{10}$$

Comme (10) est exempt d'étiquettes de régression, cette perte peut être appliquée simultanément sur les domaines source étiqueté et cible non étiqueté pendant l'entraînement. La perte d'entropie croisée conventionnelle dans (7) ne peut être utilisée que sur le domaine source en raison de la nécessité de définir le rayon $\mathbf{r}$ pour spécifier les paires positives et négatives. Cela donne à la perte simplifiée un avantage car elle crée un meilleur espace latent qui correspond à l'espace contrastif des domaines source et cible. Par conséquent, la perte contrastive simple peut être étendue comme suit :

$$\mathcal{L}_{\text{SContrast}} = \left\|\widetilde{\mathbf{F}}_{\mathcal{S}} - \mathbf{F}_{\mathcal{S}}\right\|_F + \left\|\widetilde{\mathbf{F}}_{\mathcal{T}} - \mathbf{F}_{\mathcal{T}}\right\|_F \tag{11}$$

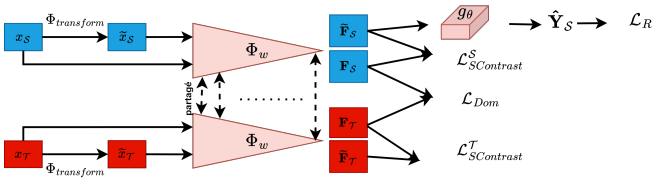

FIGURE 1 – Architecture de la méthode proposée.

TABLE 1 – Paramètres du jeu de données *dSprites*

Factor	Parameters	Task
Shape	square, ellipse, heart	recognition
Scale	$\in [0.5, 1]$	regression
Orientation	$\in [0, 2\pi]$	regression
Position X	$\in [0, 1]$	regression
Position Y	$\in [0, 1]$	regression

2.3 Adaptation de Domaine pour la Régression

L'architecture complète illustrée à la Figure 1 indique les trois fonctions objectives qui seront rétropropagées dans le réseau de neurones profond pour l'adaptation de domaine : les pertes de régression, de domaine et contrastive, selon

$$\mathcal{L}_{total} = \mathcal{L}_{\text{R}} + \alpha\,\mathcal{L}_{\text{Dom}} + \beta\,\mathcal{L}_{\text{Contrastive}},$$

où α et β contrôlent le compromis entre les trois pertes.

Pour la partie régression, les caractéristiques sont utilisées comme entrée pour un réseau de régression g_θ. Pour entraîner ce réseau, seules les données sources sont utilisées, puisque les données cibles ne sont pas étiquetées. La perte de régression est définie comme l'erreur quadratique moyenne selon

$$\mathcal{L}_{\text{R}} = \frac{1}{N_{\mathcal{S}}}\sum_{i=1}^{N_{\mathcal{S}}}\left\|y^i_{\mathcal{S}} - g_\theta\left(f^i_{\mathcal{S}}\right)\right\|^2,$$

où $g_\theta(f^i_{\mathcal{S}})$ est la valeur prédite de l'échantillon $x^i_{\mathcal{S}}$.

3 Expériences

L'évaluation se fait sur le jeu de données de référence *dSprites* [1] souvent utilisé pour les tâches de régression en adaptation de domaine. Il se compose des domaines Couleur (**C**), Bruité (**N**) et Cri (**S**), chacun de 737 280 images. Un exemple est montré dans la Figure 2 et les valeurs à prédire sont listées au Table 1. Pour notre expérience, nous visons à prédire l'échelle, la Position X et la Position Y. Le modèle est évalué sur les six tâches de transfert entre les 3 domaines.

Pour la transformation $\Phi_{transform}$, nous avons utilisé La gigue de couleurs [2] qui est une technique de transformation dans laquelle nous modifions de manière aléatoire la luminosité, le contraste, la saturation et la teinte d'une image. Ce type de transformations améliore la robustesse du modèle et le pousse à se concentrer sur la cible à l'intérieur des images. Un exemple des images transformées est montré dans la Figure 2.

1. https ://github.com/deepmind/dsprites-dataset
2. https ://pytorch.org/vision/main/generated/torchvision.transforms.ColorJitter.html

TABLE 2 – Erreur absolue moyenne des trois tâches de régression de *dSprites* (les meilleurs scores sont en rouge).

Method	C → N	C → S	N → C	N → S	S → C	S → N	Avg
ResNet-18 [11]	0.94 ± 0.06	0.90 ± 0.08	0.16 ± 0.02	0.65 ± 0.02	0.08 ± 0.01	0.26 ± 0.03	0.498
TCA [6]	0.94 ± 0.03	0.87 ± 0.02	0.19 ± 0.02	0.66 ± 0.05	0.10 ± 0.02	0.23 ± 0.04	0.498
DAN [7]	0.70 ± 0.05	0.77 ± 0.09	0.12 ± 0.03	0.50 ± 0.05	0.06 ± 0.02	0.11 ± 0.04	0.377
DANN [8]	0.47 ± 0.07	0.46 ± 0.07	0.16 ± 0.02	0.65 ± 0.05	0.05 ± 0.00	0.10 ± 0.01	0.315
JDOT [3]	0.86 ± 0.03	0.79 ± 0.02	0.19 ± 0.02	0.64 ± 0.05	0.10 ± 0.02	0.23 ± 0.04	0.468
MCD [9]	0.81 ± 0.09	0.81 ± 0.12	0.17 ± 0.12	0.65 ± 0.03	0.07 ± 0.02	0.19 ± 0.04	0.450
AFN [12]	1.00 ± 0.04	0.96 ± 0.05	0.16 ± 0.03	0.62 ± 0.04	0.08 ± 0.01	0.32 ± 0.06	0.523
RSD [1]	0.31 ± 0.03	$\mathbf{0.31 \pm 0.03}$	0.12 ± 0.02	0.53 ± 0.01	0.07 ± 0.00	0.08 ± 0.01	0.237
DL [2]	0.30 ± 0.03	0.39 ± 0.03	0.11 ± 0.02	0.44 ± 0.02	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	0.221
DL + $\mathcal{L}_{\text{EContrast}}$ (cet article)	0.30 ± 0.03	0.33 ± 0.03	0.11 ± 0.01	0.39 ± 0.03	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	0.188
DL + $\mathcal{L}_{\text{SContrast}}$ (cet article)	$\mathbf{0.28 \pm 0.03}$	0.33 ± 0.03	$\mathbf{0.08 \pm 0.01}$	$\mathbf{0.33 \pm 0.03}$	$\mathbf{0.04 \pm 0.00}$	$\mathbf{0.05 \pm 0.00}$	**0.185**

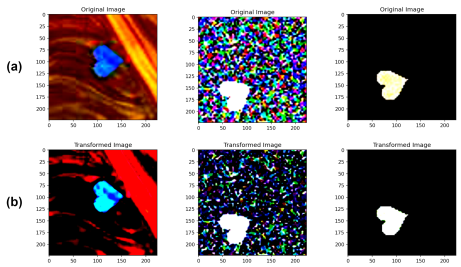

FIGURE 2 – Exemples d'images *dSprites* originales (a) et transformées via la gigue de couleurs (b).

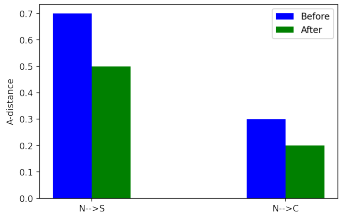

FIGURE 3 – Performances sur les tâches $\mathbf{N} \rightarrow \mathbf{C}$ et $\mathbf{N} \rightarrow \mathbf{S}$.

Le modèle a été implémenté en Pytorch et entraîné par Tesla P100. Pour le réseau d'extraction de caractéristiques, nous utilisons ResNet-18 pré-entraîné. Les étiquettes ont été normalisées à [0,1] et les images redimensionnées à 224×224 pixels. La taille des lots a été fixée à 36 et les hyperparamètres β et γ fixés à 0.001.

Pour évaluer les performances du modèle proposé, la Figure 3 montre la distance A de deux tâches de transfert, avant et après l'application du modèle. Les résultats prouvent que le changement de domaine diminue après le processus d'adaptation. La Table 1 montre une comparaison avec différentes méthodes. Le modèle affiche de meilleures performances dans presque toutes les tâches. La perte contrastive fournit une grande amélioration des résultats de la méthode DL [2], en particulier lorsque le domaine cible est le domaine (S), c'est-à-dire les tâches $\mathbf{C} \rightarrow \mathbf{S}$ et $\mathbf{N} \rightarrow \mathbf{S}$. Le domaine (S) est le domaine le plus difficile où l'objet à l'intérieur des images est ajouté à un arrière-plan peu clair. La perte contrastive aide le réseau à construire un extracteur de caractéristiques solide qui est robuste à ce bruit de fond. La complexité calculatoire est similaire à celle de la méthode DL de référence.

4 Conclusion

Dans cet article, nous avons présenté une nouvelle méthode pour l'adaptation de domaine en régression. Nous avons proposé une perte contrastive qui, combinée à la perte d'adaptation basée sur l'apprentissage du dictionnaire, permet d'améliorer les résultats de la régression. En particulier, la perte contrastive a pu bien fonctionner sur des tâches où il y a une grande confusion entre l'objet et l'arrière-plan.

Références

[1] X. Chen, S. Wang, J. Wang, and M. Long, "Representation subspace distance for domain adaptation regression," in *Proc. International Conference on Machine Learning*, pp. 1749–1759, PMLR, 2021.

[2] M. Dhaini, M. Berar, P. Honeine, and A. Van Exem, "Unsupervised domain adaptation for regression using dictionary learning," *Knowledge-Based Systems*, p. 110439, 2023.

[3] N. Courty, R. Flamary, A. Habrard, and A. Rakotomamonjy, "Joint distribution optimal transportation for domain adaptation," *Advances in Neural Information Processing Systems*, vol. 30, 2017.

[4] G. Kang, L. Jiang, Y. Yang, and A. G. Hauptmann, "Contrastive adaptation network for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4893–4902, 2019.

[5] Y. Wang, Y. Jiang, J. Li, B. Ni, W. Dai, C. Li, H. Xiong, and T. Li, "Contrastive regression for domain adaptation on gaze estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19376–19385, 2022.

[6] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang, "Domain adaptation via transfer component analysis," *IEEE transactions on neural networks*, vol. 22, no. 2, pp. 199–210, 2010.

[7] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in *International conference on machine learning*, pp. 97–105, PMLR, 2015.

[8] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The journal of machine learning research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[9] K. Saito, K. Watanabe, Y. Ushiku, and T. Harada, "Maximum classifier discrepancy for unsupervised domain adaptation," in *Proc. IEEE conference on computer vision and pattern recognition*, pp. 3723–3732, 2018.

[10] F. Wang and H. Liu, "Understanding the behaviour of contrastive loss," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2495–2504, 2021.

[11] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

[12] R. Xu, G. Li, J. Yang, and L. Lin, "Larger norm more transferable : An adaptive feature norm approach for unsupervised domain adaptation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 1426–1435, 2019.