

3DRESC-TF : Apprentissage par transfert pour la réutilisation de connaissances en classification d'objets 3D

François Meunier^{1,2}

Christophe Marsala¹

Laurent Castanié²

¹ Sorbonne Universités, UPMC Univ Paris 06, CNRS, LIP6 UMR 7606
4 place Jussieu, 75005 Paris

² Total Exploration-Production, Tour Coupole, La Défense
2 Place Jean Millier, 92800 Puteaux

Francois.Meunier@lip6.fr

Résumé

Dans cet article, nous proposons une amélioration basée sur du transfert de connaissances d'un algorithme de classification d'objets 3D existant, afin de pouvoir faire face aux manques de données, problème fréquent en apprentissage.

L'idée maîtresse réside dans le fait de transférer directement les sous-parties constructrices d'attributs d'un domaine vers un autre, afin de conserver les avantages d'interprétabilité et de souplesse de la méthode d'origine.

Mots Clefs

Classification supervisée, objets 3D, apprentissage par transfert, sélection d'attributs.

Abstract

In this article, we propose an improvement of an algorithm dealing with classification of 3D objects, using transfer learning, in order to face the lack of data, a frequent problem in learning systems.

The main idea is to directly transfer some sub-parts which are attributes builder from one domain to another in order to keep the advantages of interpretability and flexibility of the original method.

Keywords

Supervised classification, 3D objects, transfer learning, feature selection.

1 Introduction

Pour l'entreprise pétrolière Total, lors de l'étude géologique d'un sous-sol potentiellement riche en hydrocarbures, il est nécessaire de bien comprendre sa structure. C'est un élément fondamental de la chaîne de traitement de l'information en géosciences, dont la qualité et l'efficacité d'exécution se répercuteront sur le reste de l'étude. C'est en effet au cours de cette étape que le géologue peut obtenir, pour les décideurs, les éléments dont ces derniers

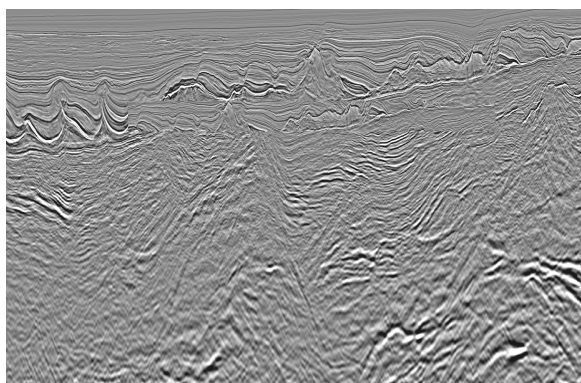


FIGURE 1 – Image sismique

ont besoin lors des choix de permis d'exploitation.

Pour cela, les géologues créent, à partir d'images sismiques reconstituées grâce à des ondes acoustiques envoyées dans le sol (Figure 1), un géo-modèle 3D censé représenter les principales couches et failles de la zone (Figure 2). Le géo-modèle, composé de surfaces 3D correspondant aux limites entre les couches sédimentaires, ne décrit que la structure géométrique du sous-sol.

Ce dernier est ensuite, en fonction des phénomènes que l'on y trouve, classé par le géologue selon les ensembles pertinents auxquels il se rattache. Les ensembles dont on parle correspondent à la présence ou à l'absence d'un certain phénomène géologique, comme par exemple des failles (normales, inverses, ...) ou des phénomènes d'interaction entre différentes couches (subduction, ...). Il s'agit donc, pour un géologue, de déterminer l'absence ou la présence de phénomènes géologiques locaux.

Jusqu'à maintenant, cette classification, comme le reste de l'étude, est réalisée manuellement par un géologue d'expérience qui connaît la zone. Cela représente une charge

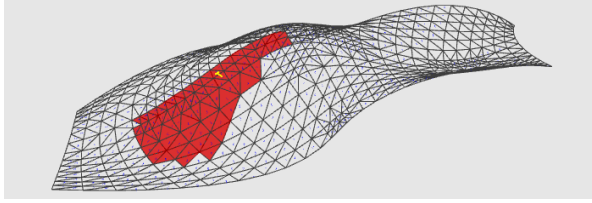


FIGURE 2 – Maillage triangulaire 3D récupéré pour chaque couche géologique

de travail conséquente, qu’une automatisation peut grandement faciliter. De plus, certains géologues disposent de moins d’expérience, tant dans le métier en lui-même que dans les personnes les plus utiles à contacter, entraînant une mauvaise évaluation des données en présence.

Néanmoins, afin de permettre une adoption par des professionnels de méthodes changeant leur manière de travailler, il est nécessaire, en plus de les convaincre par les performances des modèles présentés, de leur fournir un moyen d’appréhender de façon interprétable la méthode et/ou les résultats de cette dernière.

L’algorithme de classification supervisé de formes 3D de [Meunier et al., 2017] est une première approche du problème issue de la notion de shapelets de [Refregier, 2001], en utilisant des sous-parties discriminantes de la structure étudiée. Basé sur un système de classification supervisé traditionnelle, les résultats sont convaincants. Pourtant, il nous est apparu comme essentiel de chercher à rendre ce système moins dépendant des données spécifiques disponibles comme base d’apprentissage de la classification souhaitée. Pouvoir réutiliser d’autres classifications, dont les probabilités de distribution des données et des classes ne seraient pas trop éloignées de celle visée, est un bon moyen d’y parvenir. A cette fin, il est proposé une adaptation de cette méthode afin de pouvoir améliorer la performance du classifieur en se reposant sur de l’expérience accumulée dans d’autres contextes, ce par transfert de connaissances. Dans cet article, après une présentation de la méthode d’origine en Section 2, un état de l’art sur le transfert de connaissances est présenté en Section 3, puis en Section 4 la méthode proposée, avant que la Section 5 ne donne les perspectives de cette dernière.

2 Approche existante

En classification supervisée, certaines structures de données nécessitent un traitement particulier, impossible à réaliser par les méthodes classiques. Dans le cas des objets 3D, les méthodes actuelles ne permettent pas, sans connaissance préalable, de comprendre quelles sous-parties d’un objet peuvent être discriminantes dans le cadre de la présence du phénomène étudié, et pourquoi elles le sont. En effet, ces méthodes font le plus souvent appel à des descripteurs [Alexandre, 2012], qui extraient des particularités de l’objet, ce qui tend à caler l’objet sur

un modèle global, et donc peut ne pas tenir compte de sous-parties discriminantes.

Un certain nombre de conditions pour une nouvelle méthode se posent alors, avec notamment, pour ne citer que les plus importantes :

- comment représenter les phénomènes locaux que l’on souhaite détecter ?
- comment classifier les objets 3D ?
- comment tenir compte des sous-parties comme attributs essentiels ?
- comment permettre à l’utilisateur de comprendre le choix du classifieur ?

L’avantage évident de la méthode des shapelets appliquée aux séries temporelles est qu’il est possible de comparer des éléments de tailles très variées sans connaissance *a priori* sur ces derniers. Ce ne sont plus les éléments dans leur globalité qui peuvent être utiles, mais la juxtaposition de quelques-unes des sous-parties de ce dernier, sous-parties pertinentes possédant des caractéristiques différentes, qui, ensemble, permettent de déduire les classes d’appartenance.

L’idée maitresse est de chercher à adopter une méthode, efficace pour les séries temporelles, à un autre type de données que sont les objets 3D constitués de maillages triangulaires de points. En utilisant des descripteur locaux [Alexandre, 2012] [Dang, 2014] et des mesures de distances de répartition [Kullback and Leibler, 1951] [Matusita, 1955], les résultats s’avèrent supérieurs à l’état de l’art, et permettent en plus une visualisation des extraits les plus pertinents utilisés lors de cette classification.

Le but est de classifier les formes selon la présence ou non d’angles, le processus est représenté sur la Figure 3, et se décompose comme suit :

1. on extrait aléatoirement des sous-parties ;
2. on les trie par pertinence ;
3. on utilise les plus discriminantes pour classer de nouveaux objets.

De la méthode de [Ye and Keogh, 2009], on adapte la technique des shapelets appliquée aux séries temporelles combinée à l’approche de [Renard et al., 2015] dans laquelle la sélection aléatoire de sous-parties potentiellement discriminantes est effectuée (sélectionner, pour ces tests, l’ensemble des candidats possibles devient bien trop long en termes de temps de calcul).

Dans l’algorithme 1, nommé 3DRESC pour *3D Random Extraction of Sub-parts for Classification*, on a :

- *Extraction(objet, tailles, nombres)* qui extrait d’un objet 3D un certain nombre de sous-parties de tailles prédéfinies ;
- *CalculDescripteur(sous – sur face)* qui calcule le descripteur pour chaque point de la sous-surface ;

Algorithme 1 : 3DRESC [Meunier et al., 2017]

```

1  Entrées :
2  Objets : objets 3D; Classes : classes correspondantes;
   k : nombre d'attributs souhaité; tailles : tailles des
   extraits; nombres : nombre d'extraits pour chaque taille
   ;
3  Sorties :
4  SousParties : Liste de sous-parties discriminantes
   d'objets 3D ;
5  Début
6  i ← 0;
7  pour Objet dans Objets faire
8  |   i ← i + 1;
9  |   j ← 0;
10 |   candidats ←
11 |   |   Extraction(Objet, tailles, nombres);
12 |   |   pour candidat dans candidats faire
13 |   |   |   j ← j + 1;
14 |   |   |   DescObjet3D[i][j] ←
15 |   |   |   |   CalculDescripteur(candidat);
16 |   |   fin
17 |   fin
18 |   obj1 ← 0;
19 |   pour Objet1 dans Objets faire
20 |   |   obj1 ← obj1 + 1;
21 |   |   obj2 ← 0;
22 |   |   pour Objet2 dans Objets faire
23 |   |   |   obj2 ← obj2 + 1;
24 |   |   |   si Objet1 différent de Objet2 alors
25 |   |   |   |   Distances ← min(CalculDistances(
26 |   |   |   |   |   DescObjet3D[obj1], DescObjet3D[obj2]));
27 |   |   |   |   Gains ←
28 |   |   |   |   |   CalculGains(Distances, Classes);
29 |   |   |   |   |   ListeGain[i] ← max(Gains);
30 |   |   |   fin
31 |   |   fin
32 |   ListeGain ← Ordonner(ListeGain);
33 |   SousParties ← ListeGain[1, ..., k];
34 Fin

```

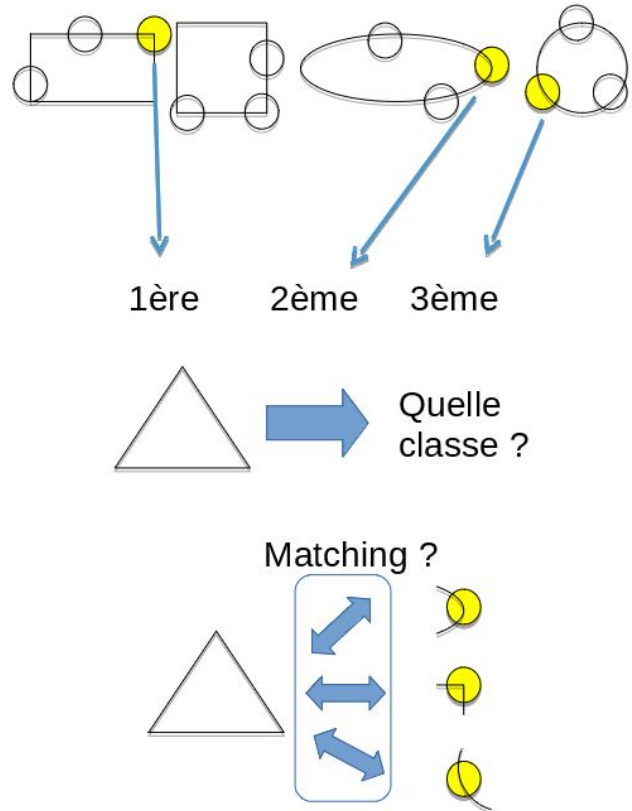


FIGURE 3 – Classification des formes

- *CalculDistances*(*Desc1*, *Desc2*) qui calcule, à partir de deux descripteurs de 2 objets, la similitude de leurs histogrammes de répartition de valeurs.
- *CalculGains*(*distances*, *classes*) qui calcule par la formule *f-stat* le gain d'information de l'attribut (créé par une sous-surface particulière) correspondant pour la classification souhaitée.

En présence de données en quantité suffisante, cette méthode s'est avérée utile aux géologues, que ce soit pour la classification en elle-même que pour la compréhension des zones choisies car pertinentes. Les performances des techniques de l'état de l'art ont bien été surpassées, et l'interprétabilité par la visualisation de données pertinentes a permis de plus facilement intégrer la méthode à la chaîne de traitement existante en géosciences.

3 L'apprentissage par transfert

Une hypothèse des méthodes traditionnelles d'apprentissage automatique est que les données d'apprentissage et les données de test sont issues du même domaine, de sorte que l'espace des attributs en entrée et la distribution des données sont les mêmes. Cependant, dans beaucoup de scénarios d'apprentissage supervisé, cette hypothèse ne

tient pas.

Il est donc nécessaire de créer des méthodes d'apprentissage performantes formées à partir de données plus facilement récupérables, voir simplement les seules disponibles. Ce concept est appelé apprentissage par transfert, ou *transfer learning* [Dai et al., 2007] [Lu et al., 2015], et est illustré par la Figure 4.

Pour un domaine d'apprentissage particulier, si l'on considère les notations de [Pan and Yang, 2010], alors :

- $D = \{\chi, P(X)\}$, avec χ l'espace d'attributs et $X = \{x_1, x_2, \dots, x_n\} \in \chi$;
- $T = \{\Upsilon, f(\cdot)\}$ avec Υ un espace de *labels*, et $f(\cdot)$ une fonction de prédiction.

Nous pouvons par ailleurs décomposer l'apprentissage par transfert en deux ensembles, que sont, selon [Weiss et al., 2016] :

1. Si $X_S = X_T$ alors il est défini comme un apprentissage par transfert homogène. La majorité des solutions d'apprentissage par transfert homogène utilisent l'une des trois principales stratégies qui consistent à corriger, entre les données source et cible, la différence de distribution marginale, la différence de distribution conditionnelle, ou à la fois les différences de distribution marginale et conditionnelle.
2. Sinon, on parle d'apprentissage par transfert hétérogène. La majorité des solutions d'apprentissage par transfert hétérogènes sont axées sur l'alignement des espaces d'entrée des domaines source et cible, avec comme hypothèse que les distributions de domaines restent les mêmes.

Dans notre domaine, et dans ces deux cas, la principale façon de procéder reste l'extraction d'un modèle de cartographie des éléments ou des distributions d'un ou plusieurs domaines (appelé(s) domaine(s) source(s)) vers un autre (appelé domaine cible). Ce faisant, nous perdons notre capacité initiale qui consistait à pouvoir visualiser les raisons qui ont amené notre algorithme à donner un résultat particulier.

4 L'adaptation de la méthode initiale à un contexte de transfert de connaissances

4.1 Description de l'approche

Supposons que l'on dispose d'un domaine source unique avec beaucoup d'exemples d'apprentissage (en fait une tâche de classification réalisée sur des données particulières), et un domaine cible avec seulement quelques données déjà annotées (dans notre cas, environ 25% du nombre total des objets dans le domaine cible). Ce cas d'étude est illustré sur la Figure 4.1. En utilisant la méthode

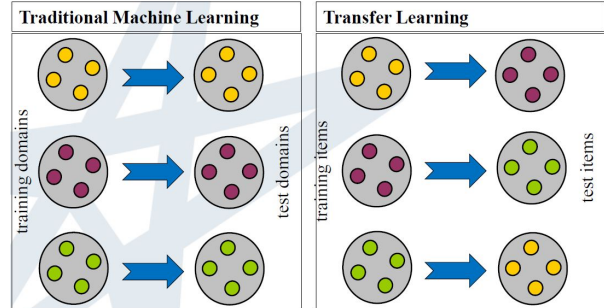


FIGURE 4 – Une définition intuitive du transfert learning - [Pan and Yang, 2010]

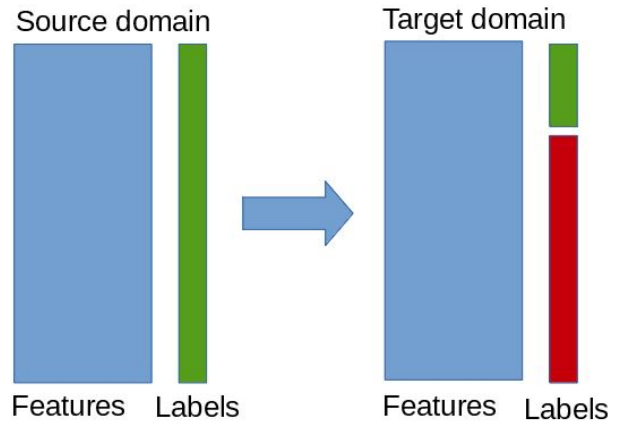


FIGURE 5 – Notre problème d'apprentissage par transfert : en vert, les éléments labellisés, et en rouge ceux qui ne le sont pas

précédemment proposée par [Meunier et al., 2017], nous n'apprenons que sur les quelques exemples disponibles du domaine cible et prédisons le reste. Cependant, avec le transfert de connaissances, le domaine source devient précieux, car il permettrait de sélectionner également les sous-parties qui peuvent rester pertinentes en passant du domaine source vers le domaine cible et de les ajouter aux sous-parties déjà sélectionnées grâce la méthode précédente (donc exclusivement issues du domaine cible). C'est la raison pour laquelle il est proposé de transférer simplement certaines des sous-parties les plus pertinentes extraites d'un domaine vers un autre, après avoir vérifié que la pertinence de cette sous-partie dans le domaine cible est bonne. Ces sous-parties transférées restent les constructeurs d'attributs sur lesquels l'apprentissage est réalisé.

Le processus se déroule comme suit :

1. tout d'abord, la méthode classique de sélection des sous-parties pertinentes dans le domaine cible est réalisée ;
2. parallèlement, la même chose est faite pour le domaine source ;
3. ensuite, les sous-parties pertinentes du domaine source sont testées au sein du domaine cible afin d'évaluer leur capacité à y être réutilisées ;
4. enfin, les sous-parties de l'extraction originale du domaine cible et les sous-parties choisies dans le domaine source sont toutes deux utilisées pour classer l'ensemble de données du le domaine cible.

Soient O_1, \dots, O_n l'ensemble des objets labellisés qui constituent notre base d'apprentissage :

- $\forall O_{i=1, \dots, n}$, on extrait aléatoirement m sous-surfaces (pour chaque objet O_i) que sont $S_{i,1}, \dots, S_{i,m}$.
- On considère un descripteur d'objet 3D particulier $Desc$, et une distance $Dist$ (qui peut être une distance entre répartition de valeurs, pour le cas des descripteurs d'objets). $\forall i \in (1, \dots, n)$ et $\forall j \in (1, \dots, m)$, on calcule le descripteur $Desc(S_{i,j})$ de la sous-surface $S_{i,j}$.

Puis, pour chaque sous-surface $S_{i,j}$ à évaluer :

- $\forall i' \in (1, \dots, n) \setminus i$ et $\forall j' \in (1, \dots, m)$, on calcule la proximité définie comme

$$Prox(S_{i,j}, S_{i',j'}) = Dist(Desc(S_{i,j}), Desc(S_{i',j'}))$$

- Les degrés correspondant à ses valeurs d'imbrication dans chacun des objets O_1, \dots, O_n valent : $\forall i' \in (1, \dots, n)$,

$$Degré(s_{i,j}, O_{i'}) = \inf_{l=1, \dots, m} (Prox(S_{i,j}, S_{i',l}))$$

Ce degré de matchage ou d'appartenance d'une sous-surface à un objet permet d'établir une forme de proximité entre la sous-surface étudiée et chaque objet. C'est à partir de ces valeurs que le score de pertinence de chacune des sous-surfaces est évalué, afin de ne sélectionner que les plus discriminantes.

Ce score, proposé la première fois par [Lines et al., 2012], et dont la pertinence pour le domaine étudié a été confirmée par [Meunier et al., 2017], est la mesure de $f-stat$:

$$f-stat = \frac{\frac{1}{C-1} \sum_{i=1}^C (\bar{D}_i - \bar{D})^2}{\frac{1}{n-C} \sum_{i=1}^C \sum_{d_j \in D_i} (d_j - \bar{D}_i)^2} \quad (1)$$

avec $C > 1$ le nombre de classes, n le nombre de sous-parties ($n > C$), \bar{D}_i la moyenne des degrés entre les surfaces de classe i et la sous-partie concernée, et \bar{D} la moyenne des degrés entre la sous-partie et les objets 3D. Par la suite, les k premiers candidats selon ce critère sont utilisés pour le calcul des attributs, k est défini expérimentalement en moyenne à environ 0.1% du nombre total d'extraits.

Dans notre cas, pour le domaine source, tous les extraits ne seront pas conservés tels quels pour la construction d'attributs, il est en effet d'abord nécessaire de les tester pour s'assurer de leur pertinence. La valeur $f-stat$ calculée pour chaque sous-surface du domaine source au sein du domaine cible doit être au minimum du même ordre de grandeur que les extraits du domaine cible qui ont été sélectionnés.

Pour chaque extrait dont le test est positif, ce dernier est ajouté au domaine cible (avec ses propres extraits), et le transfert se base donc sur l'extrait constructeur d'attribut, et non pas l'attribut lui-même.

Soit O_S l'ensemble des sous-parties sélectionnées au sein du domaine source, O_C ceux du domaine cible, et $Transfert_{Extrait}$ une fonction de seuil qui prend comme arguments une sous-surface extraite et un domaine possédant une classification. La fonction renvoie une réponse positive si l'extrait a une pertinence (fonction $f-stat$) supérieure au seuil choisi, et est donc ajouté (transféré) au extraits O_C , et sera ignoré sinon.

Avec, dans l'algorithme 2 :

- $CapacitSeparation$, fonction qui calcule le score d'une sous-partie selon le test $f-stat$;
- $Seuil_{TF}$, le seuil défini empiriquement comme étant le minimum de pertinence de sous-parties sélectionnées de l'ensemble d'entraînement, et la capacité de séparation utilisée reste la même que dans l'algorithme d'origine, c'est-à-dire le $f-stat$.

Algorithme 2 : Transfert d’extraits pertinents

```
1 Entrées :  
2  $Objets_{Source}$  : objets 3D du domaine source ;  
    $Objets_{Cible}$  : objets 3D du domaine cible ;  
    $SousParties$  : sous-parties issues du domaine source ;  
    $Classes$  : classes correspondantes pour chaque objet ;  
    $Seuil_{TF}$  : limite arbitraire de pertinence pour le transfert  
   des sous-parties du domaine source ;  
3 Output :  
4  $SousPartiesPertinentes$  : Liste de sous-parties  
   pertinentes d’objets 3D utilisable pour le transfer  
   learning, donc venant du domaine source ;  
5 Début  
6  $SousPartiesPertinentes_{Transferees} \leftarrow 0$  ;  
7 pour  $SousPartie \in SousParties$  faire  
8    $Pertinence \leftarrow$   
    $CapacitSeparation(SousPartie, Classes, Objets_{Cible})$  ;  
9   si  $Pertinence \geq Seuil_{TF}$  alors  
10     $SousPartiesPertinentes_{Transferees} \leftarrow$   
     $SousPartiesPertinentes \cup SousPartie$  ;  
11 fin  
12 Fin
```

Les sous-parties finalement utilisées pour la construction des attributs sont l’union de celles issues du domaine cible et celles sélectionnées au sein du domaine source qui sont aussi pertinentes dans le domaine cible, à savoir :

$$SousParties_{Cible} \cup SousParties_{Pertinentes_{Transferees}}$$

De cette manière, nous obtenons des résultats qui comparent les approches en termes de performances. Ils ont été rassemblés à partir de tests multiples en utilisant la sélection aléatoire des objets, les seules contraintes étaient d’avoir deux classes dans chaque domaine.

L’algorithme global correspondant est *3DRESC-TF*, pour *3D Random Extraction of Sub-parts for Classification with Transfer Learning*.

4.2 Expérimentations

3DRESC-TF est testé avec des données libres du *Princeton Shape Benchmark*¹ (PSB), dont les données proviennent de multiples ensembles d’objets 3D déjà existants. Ces derniers sont ceux qui, en termes de formats et de types, se rapprochent le plus des géo-modèles 3D dont nous disposons en entreprise. Les classes les plus fines sont retenues (3 niveaux étant possibles), afin de rajouter encore davantage de contraintes et de difficultés à la prédiction lors des tests.

Le classifieur utilisé une fois les attributs récupérés est celui des forêts aléatoires du package Python *scikit-learn*. Les tests sont effectués sur un processeur *Intel Core i7 vPro* avec 16 Go de RAM.

1. <http://shape.cs.princeton.edu/benchmark/> consulté le 05/02/2017

Ainsi, nous réutilisons la connaissance de plusieurs domaines disponibles et pertinents dans le but d’améliorer notre prédiction. Dans le cas de domaines très différents, ce transfert d’apprentissage pourrait faire face à un léger apprentissage dit négatif (baisse de la performance en raison d’informations transmises mais non pertinentes, voir nuisibles), et donc devrait logiquement rester inutilisées, ce qui reviendrait à n’employer que l’algorithme d’origine.

Face au potentiel déséquilibre de classe au sein de ces nouvelles expériences, nous choisissons d’utiliser la *AUC* (aire sous la courbe ROC - *Receiver Operating Characteristic*) pour évaluer les performances du modèle.

La première chose à souligner, et qui s’avère intuitive, est le fait que plus le nombre d’éléments dans le domaine cible est faible, plus notre méthode d’apprentissage par transfert sera bénéfique vis-à-vis du processus d’apprentissage classique. En effet, moins d’exemples signifie un apprentissage plus difficile, et donc l’intérêt pour le transfert des connaissances venant d’autres domaines augmente fortement. Nous gardons néanmoins l’avantage de la méthode originale qui est d’être en mesure d’apprendre sur un jeu de données restreint.

Cependant, on peut mentionner le fait qu’en ajoutant simplement de nouvelles fonctionnalités du domaine source, on améliorerait statistiquement la performance si on utilise un bon algorithme de classification (forêts aléatoires, par exemple).

Pour supprimer ce possible biais dans nos résultats et pouvoir observer le seul impact de notre proposition d’apprentissage de transfert, nous avons tout simplement décidé de comparer notre méthode d’apprentissage mécanique classique en utilisant les 200 extraits par objet plus le nombre de sous-parties transférées possibles dans l’apprentissage par transfert, avec l’approche d’apprentissage par transfert elle-même. Ainsi, les résultats de la Table 1 confirment que cet algorithme d’apprentissage par transfert améliore clairement la manière classique de calculer notre méthode.

Ensuite, et toujours en essayant de donner à l’utilisateur une explication des résultats donnés, nous proposons de faire une visualisation d’une tâche d’apprentissage par transfert particulière.

A partir de deux classifications d’avions représentés sous forme de maillages 3D, la première étant le domaine source et la seconde le cible, nous donnons le nombre d’éléments utilisés comme apprentissage dans le domaine cible et montrons les sous-parties les plus pertinentes (transférées et du domaine cible).

La première classification, c’est-à-dire le domaine source, vise à séparer le biplan des avions commerciaux, tandis que la seconde différencie les chasseurs des planeurs.

En utilisant ces deux domaines pour la visualisation, nous obtenons finalement les sous-parties suivantes comme les plus pertinentes extraites du domaine source et réutilisées

TABLE 1 – Classification en fonction du nombre d’objets pour les domaines cible et source

Nombre d’objets du domaine source	Nombre d’objets du domaine cible	Sans apprentissage par transfert	Avec apprentissage par transfert
60	40	0.616 ± 0.037	0.642 ± 0.034
	30	0.581 ± 0.049	0.617 ± 0.047
	20	0.545 ± 0.055	0.583 ± 0.065
80	40	0.611 ± 0.035	0.666 ± 0.036
	30	0.580 ± 0.041	0.625 ± 0.041
	20	0.548 ± 0.054	0.601 ± 0.060
100	40	0.616 ± 0.032	0.640 ± 0.034
	30	0.579 ± 0.045	0.615 ± 0.048
	20	0.544 ± 0.058	0.581 ± 0.069

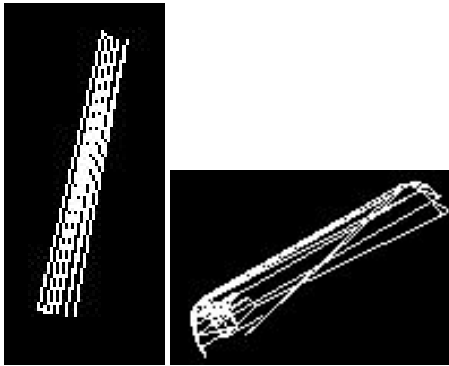


FIGURE 6 – A gauche, l’aile d’un planeur et à droite, le cockpit d’un biplan

au sein de la cible.

Dans un cas particulier, à partir des formes d’un planeur et d’un biplan, les sous-parties récupérées pour le transfert car pertinentes sont données dans la Figure 6.

Nous remarquons que la première sous-partie représente des ailes très minces, et il est logique de la transférer, car ces ailes très particulières sont caractéristiques d’un planeur (longues et minces), et si on les compare à un chasseur à réaction, permettent d’améliorer la classification en utilisant ce critère.

Le second est un peu plus centré sur le corps de l’avion. On peut aisément reconnaître un cockpit de biplan, qui apporte une connaissance sur les moyens utilisés par le classifieur pour reconnaître la classe d’appartenance.

Testée sur des données réelles de Total, la méthode a encore besoin, pour être validée, de voir son intérêt confirmé par le métier, que la mise en pré-production devrait permettre d’ici peu.

5 Conclusions et perspectives

Dans cet article, nous avons proposé un nouvel algorithme, appelé 3DRESC-TF, qui vient compléter de précédents tra-

vaux sur la classification d’objets 3D. Afin de faire face à un manque de données d’apprentissage, cette adaptation vise à transférer de la connaissance issue d’autres apprentissages, dans le but d’améliorer le taux de classification. Proposant des performances supérieures aux méthodes de l’état de l’art, nous valorisons de cette manière les apprentissages réalisés par le passé, afin, d’une part, d’améliorer la classification en cours, et d’autre part de pouvoir faire face à des données différentes de celles utilisées pour l’apprentissage.

Néanmoins, les sous-parties extraites et récupérées pour la visualisation ne sont à l’heure actuelle pas beaucoup plus interprétables avec cette méthode que celles obtenues par la méthode d’origine, et la pertinence d’une utilisation systématique de cette nouvelle fonctionnalité devra être, à l’avenir, étudiée pour le cas d’application concrète qui est le notre à Total.

Références

- [Alexandre, 2012] Alexandre, L. A. (2012). 3D Descriptors for Object and Category Recognition : a Comparative Evaluation. In *Workshop on Color-Depth Camera Fusion in Robotics at the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Vilamoura, Portugal.
- [Dai et al., 2007] Dai, W., Yang, Q., Xue, G.-R., and Yu, Y. (2007). Boosting for transfer learning. In *Proceedings of the 24th International Conference on Machine Learning, ICML ’07*, pages 193–200, New York, NY, USA. ACM.
- [Dang, 2014] Dang, Q. V. (2014). *Similarités dans des modèles BRep paramétriques : detection et applications*. PhD thesis, Université de Toulouse.
- [Kullback and Leibler, 1951] Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1) :79–86.
- [Lines et al., 2012] Lines, J., Davis, L. M., Hills, J., and Bagnall, A. (2012). *A Shapelet Transform for Time Series Classification*. KDD ’12. ACM, New York, NY, USA.

- [Lu et al., 2015] Lu, J., Behbood, V., Hao, P., Zuo, H., Xue, S., and Zhang, G. (2015). Transfer learning using computational intelligence : A survey. *Knowledge-Based Systems*, 80 :14 – 23. 25th anniversary of Knowledge-Based Systems.
- [Matusita, 1955] Matusita, K. (1955). Decision rules, based on the distance, for problems of fit, two samples, and estimation. *Annals of Mathematical Statistics*, 26(4) :631–640.
- [Meunier et al., 2017] Meunier, F., Marsala, C., Castanié, L., and Conche, B. (2017). Classification d’objets 3d par extraction aléatoire de sous-parties discriminantes pour l’étude du sous-sol en prospection pétrolière. In *Extraction et Gestion des Connaissances*, pages 225–236.
- [Pan and Yang, 2010] Pan, S. J. and Yang, Q. (2010). A survey on transfer learning. *IEEE Trans. on Knowl. and Data Eng.*, 22(10) :1345–1359.
- [Refregier, 2001] Refregier, A. (2001). Shapelets : I. a method for image analysis. *Mon. Not. Roy. Astron. Soc.*, 338 :35.
- [Renard et al., 2015] Renard, X., Rifqi, M., and Detyniecki, M. (2015). Random-shapelet : an algorithm for fast shapelet discovery. In *Proceedings of the IEEE International Conference on Data Science and Advanced Analytics (DSAA’2015)*, pages 1–10.
- [Weiss et al., 2016] Weiss, K., Khoshgoftaar, T. M., and Wang, D. (2016). A survey of transfer learning. *Journal of Big Data*, 3(1) :9.
- [Ye and Keogh, 2009] Ye, L. and Keogh, E. (2009). Time Series Shapelets : A New Primitive for Data Mining. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD ’09*, pages 947–956, New York, NY, USA. ACM.